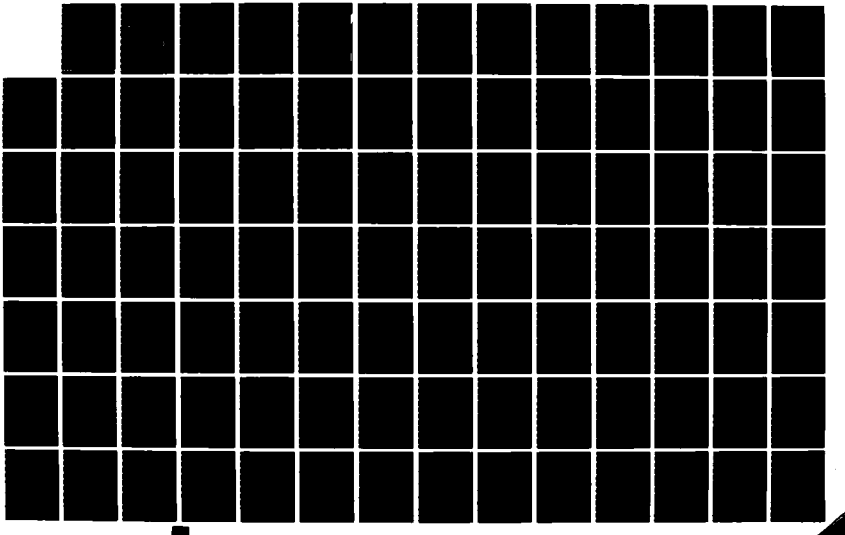
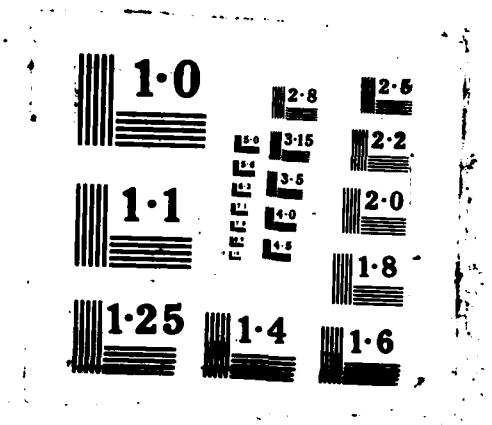
AD-A176 064 ROUND ROBIN SCHEDULING FOR FAIR FLOW CONTROL IN DATA COMMUNICATION NETWOR..(U) MASSACHUSETTS INST OF TECH CAMBRIDGE LAB FOR INFORMATION AND D.. E L HAHNE DEC 86 1/3

UNCLASSIFIED LIDS-TH-1631 N00014-84-K-0357 F/G 17/2 NL




1·0
1·1
1·25
1·4
1·6
1·8
2·0
2·2
2·5
2·8
3·15
3·5
4·0
4·5
5·0

(12)

DECEMBER 1986

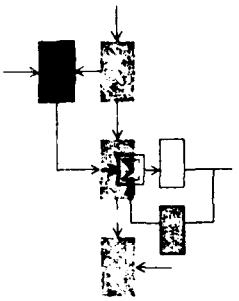
LIDS-TH-1631

Research Supported By:

Defense Advanced Research Projects Agency
ONR/N00014-84-K-0357

Army Research Office
DAAG-29-84-K-0005

National Science Foundation
NSF-ECS-8310698



ROUND ROBIN SCHEDULING FOR FAIR FLOW CONTROL IN DATA COMMUNICATION NETWORKS

Ellen Louise Hahne

Laboratory for Information and Decision Systems

MASSACHUSETTS INSTITUTE OF TECHNOLOGY, CAMBRIDGE, MASSACHUSETTS 02139

December 1986 LIDS-TH-1631

ROUND ROBIN SCHEDULING FOR FAIR FLOW CONTROL IN DATA COMMUNICATION NETWORKS

by

Ellen Louise Hahne

This report is based on the unaltered thesis of Ellen Louise Hahne submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at the Massachusetts Institute of Technology in December of 1986. This research was conducted at the M.I.T. Laboratory for Information and Decision Systems with partial support provided by the Defense Advanced Research Projects Agency under Contract ONR/N00014-84-K-0357, the Army Research Office under Contract DAAG29-84-K-0005, and the National Science Foundation under Grant NSF-ECS-8310698.

Laboratory for Information and Decision Systems
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered) UNCLASSIFIED

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. ADA176064	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) ROUND ROBIN SCHEDULING FOR FAIR FLOW CONTROL IN DATA COMMUNICATION NETWORKS	5. TYPE OF REPORT & PERIOD COVERED Thesis
	6. PERFORMING ORG. REPORT NUMBER LIDS-TH-1631
7. AUTHOR(s) Ellen Louise Hahne	8. CONTRACT OR GRANT NUMBER(s) DARPA Order No. 3045/2-2-84 Amendment #11 ONR/N00014-84-K-0357
9. PERFORMING ORGANIZATION NAME AND ADDRESS Massachusetts Institute of Technology Laboratory for Information and Decision Systems Cambridge, Massachusetts 02139	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Program Code No. 5T10 ONR Identifying No. 049-383
11. CONTROLLING OFFICE NAME AND ADDRESS Defense Advanced Research Projects Agency 1400 Wilson Boulevard Arlington, Virginia 22209	12. REPORT DATE December 1986
	13. NUMBER OF PAGES 222
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Office of Naval Research Information Systems Program Code 437 Arlington, Virginia 22217	15. SECURITY CLASS. (of this report) UNCLASSIFIED
	15a. DECLASSIFICATION/ DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release: distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

round robin	fairness	computer communication	schedilling
cyclic service	throughput	packet switching	optimization
polling	delay	network	
window	integrated services	queueing	
flow control	data communication	traffic	

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This thesis studies a simple strategy for fairly allocating link capacity in a point-to-point packet network with virtual circuit routing. Each link offers its packet transmission slots to its user sessions by polling them in round robin order. In addition, link-by-link window flow control is used to prevent excessive packet queues at the network nodes. As the window size increases, the session throughput rates are shown to approach limits that are perfectly fair in the max-min sense. That is, the smallest session rate in the network is as large as possible and, subject to that constraint, the second-smallest session rate is as

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-LF-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. (Continued)

large as possible, etc. If each session has evenly spaced packet arrivals or has such heavy demand that packets are always waiting to enter the network, then a finite window size suffices to produce perfectly fair throughput rates. (These properties do not hold if first-come-first-served scheduling is used instead of round robin.)

The round robin method is considerably simpler than other known strategies for achieving throughput fairness. The fair session rates are not explicitly computed, and the only overhead communication is that required for the window acknowledgements. The main drawback is that large windows are needed to achieve even approximately fair throughputs in some systems, and large windows permit large cross-network delays. This may be intolerable for some users. However, the thesis also shows that if a session elects to use small windows, its packets are guaranteed to experience small cross-network delays, and a certain lower bound on its service rate is still guaranteed. (This service rate determines the maximum session throughput rate that can be supported and also roughly determines, for a given throughput rate, the delay of packets waiting to be admitted to the network.) These guarantees for sessions with small windows apply even if other sessions in the network are using larger windows. Thus the round robin method seems to be well suited to integrated services networks. Delay-sensitive sessions can use small windows to meet their needs, and the remaining transmission capacity can be fairly divided among the other sessions by assigning them large windows. Moreover, a session with throughput and/or delay requirements too stringent to be met simply by proper window sizing could be given priority service by being visited more than once in each polling cycle.

ROUND ROBIN SCHEDULING FOR FAIR FLOW CONTROL IN DATA COMMUNICATION NETWORKS

by

ELLEN LOUISE HAHNE

B.S., Rice University, 1978
S.M., Massachusetts Institute of Technology, 1981
E.E., Massachusetts Institute of Technology, 1984

Submitted in Partial Fulfillment
of the Requirements for the Degree
of
DOCTOR OF PHILOSOPHY
in
ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
February 1987

Signature of Author Ellen L. Hahne
Dept. of Electrical Engr. and Computer Sci.
December 9 , 1986

Certified by Robert G. Gallager
Professor Robert G. Gallager
Thesis Supervisor

Accepted by _____
Professor Arthur C. Smith, Chairman
Dept. Committee on Graduate Students

ROUND ROBIN SCHEDULING FOR FAIR FLOW CONTROL IN DATA COMMUNICATION NETWORKS

by

ELLEN LOUISE HAHNE

Submitted to the Department
of Electrical Engineering and Computer Science
on December 9, 1986
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy

ABSTRACT

This thesis studies a simple strategy for fairly allocating link capacity in a point-to-point packet network with virtual circuit routing. Each link offers its packet transmission slots to its user sessions by polling them in round robin order. In addition, link-by-link window flow control is used to prevent excessive packet queues at the network nodes. As the window size increases, the session throughput rates are shown to approach limits that are perfectly fair in the max-min sense. That is, the smallest session rate in the network is as large as possible and, subject to that constraint, the second-smallest session rate is as large as possible, etc. If each session has evenly spaced packet arrivals or has such heavy demand that packets are always waiting to enter the network, then a finite window size suffices to produce perfectly fair throughput rates. (These properties do not hold if first-come-first-served scheduling is used instead of round robin.)

The round robin method is considerably simpler than other known strategies for achieving throughput fairness. The fair session rates are not explicitly computed, and the only overhead communication is that required for the window acknowledgments. The main drawback is that large windows are needed to achieve even approximately fair throughputs in some systems, and large windows permit large cross-network delays. This may be intolerable for

some users. However, the thesis also shows that if a session elects to use small windows, its packets are guaranteed to experience small cross-network delays, and a certain lower bound on its service rate is still guaranteed. (This service rate determines the maximum session throughput rate that can be supported and also roughly determines, for a given throughput rate, the delay of packets waiting to be admitted to the network.) These guarantees for sessions with small windows apply even if other sessions in the network are using larger windows. Thus the round robin method seems to be well suited to integrated services networks. Delay-sensitive sessions can use small windows to meet their needs, and the remaining transmission capacity can be fairly divided among the other sessions by assigning them large windows. Moreover, a session with throughput and/or delay requirements too stringent to be met simply by proper window sizing could be given priority service by being visited more than once in each polling cycle.

Keywords: round robin, cyclic service, polling, window, flow control, fairness, throughput, delay, integrated services, data communication, computer communication, packet switching, network, queuing, traffic, scheduling, optimization.

Thesis Supervisor: Robert G. Gallager
Title: Professor of Electrical Engineering and Computer Science

ACKNOWLEDGMENTS

I acknowledge with pleasure and gratitude the many contributions of my thesis advisor Prof. Robert Gallager. He taught me how to think about data networks. The thesis topic was his suggestion. In the course of dozens of meetings, he made many important technical contributions. His outrageous optimism overcame "insurmountable" technical barriers. He read the thesis with extreme care ·ot resting until he found the typo in Section 4.5.2, and he taught me the ce between "which" and "that." By arranging most of my financi· .·, he freed me to waste my time in more creative ways.

This work has also benefited from my discussions with Dr. Utpal Mukherji, whose thesis on a similar topic was a great influence, Profs. Dimitri Bertsekas and Thomas Magnanti, who served as thesis readers, Prof. Pierre Humblet, and Drs. Bharath Kumar Kadaba, Jeannine Mosely, Jean Regnier, and Isidro Castineyra.

Financial support was provided by grants from the Defense Advanced Research Projects Agency (Contract ONR/N00014-84-K-0357), the Army Research Office (Contract DAAG29-84-K-0005), and the National Science Foundation (Grant NSF-ECS-8310698), by a Vinton Hayes Fellowship in Communications, and by an AT&T Bell Laboratories Doctoral Scholarship. I thank my sponsors for their generous support.

I am indebted to Mr. Orlando Sotomayor-Diaz for UNIX advice and for massive aid in producing this document. (Alas, he is already thinking of ways I might repay him.) Ms. Nancy Young was most helpful in arranging countless meetings and mailings.

For behind-the-scenes emotional support, I thank my family, my officemates (too numerous to list after four years and three offices), my pals Orlando Sotomayor-Diaz, Isidro Castineyra, Jennifer Lundelius Welch, Utpal Mukherji, and John Spinelli, and my mentors Stan Gershwin, Al Drake, Bob Gallager, and Dimitri Bertsekas. I sincerely thank these last three for all the advice I rejected.

This thesis is dedicated to Al Drake, who encouraged me to start it, and to Orlando Sotomayor-Diaz, who encouraged me to finish it.

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Problem Statement and Background

Consider a data communication network consisting of store-and-forward switches (*nodes*) joined by point-to-point communication channels (*links*). Each network user (*session*) is assigned a fixed path (*virtual circuit*) through the network, and data for the session are sent in manageable parcels (*packets*) along this path. In such a network, occasional surges in user demand can overload network links, causing packet queues to build up in network nodes. These queues may eventually overflow the nodes' storage space, or the delay of acknowledgments may cause transmitters to assume that data were lost. These problems result in wasteful retransmissions that effectively reduce the capacity of the network. Flow control procedures attempt to prevent or alleviate this degradation by regulating the appropriate traffic sources. Gerla and Kleinrock [10] discuss many of the flow control techniques that have been proposed in the literature.

One such scheme is the *window method* [10]. This technique limits the number of packets for each session that have been transmitted but for which acknowledgments have not yet been received. The maximum permissible number of outstanding packets is called the *window size*. In the *end-to-end* method, a single window is applied to all of a session's traffic, and the session's destination node sends an acknowledgment to the origin node whenever a packet is claimed by the session's sink. In the *link-by-link* or *node-by-node*

method, the session has a separate window for its traffic over each link, and whenever a packet is transmitted from a node, that node sends an acknowledgment to the packet's preceding node. The link-by-link method is equivalent to each session having a dedicated storage area (*buffer*) at each node in its path; the buffer capacity equals the window size. The window method is described here because it is a component of several more elaborate strategies to be discussed later.

It would be desirable for flow control procedures to regulate network inputs so as to grant each session a fair throughput rate. Gerla and Kleinrock [10] explain that many proposed flow control methods are unfair. Several studies have addressed the issue of throughput fairness, however, and these will now be briefly discussed.

The problem of achieving fair throughput rates can be broken into three parts. First the fairness objective must be formulated precisely. Then the fair session rates must be determined. Finally, these rates must be enforced. Hayden [13: Chapter 3], Regnier [23], Golestaani and Gallager [12, 8], Gerla and Staskauskas [11: Section 5.2], Thaker and Cain [26], Ibe [14], Gafni [5: Sections 4 and 6.2], Sauve, Wong and Field [24, 25], and Bharath-Kumar and Jaffe [1, 18] have objectives of roughly the same form. They seek to maximize a sum of functions, one for each session. For Hayden, each term gives the satisfaction of a session as a function of its throughput rate. Regnier considers both the throughput rate of a session and its average packet delay. Golestaani and

Gallager, Gerla and Staskauskas, Thaker and Cain, Ibe, and Gafni express a session's happiness as a function of only its throughput rate, but extra terms are added to penalize high link delays. Sauve, Wong and Field use a performance measure that depends on the ratio of a session's throughput rate to the total network throughput rate. Bharath-Kumar and Jaffe measure a session's success by its *power* (i.e., throughput divided by delay) or the logarithm of its power. Another fairness approach, called *max-min flow control* or *bottleneck flow control*, is used in various forms by Bially, Gold, and Seneff [2], Jaffe [16, 17], Gerla and Staskauskas [11: Section 3], Hayden [13: Chapter 4], Gafni [5: Chapter 3], Gafni and Bertsekas [6], Oshinsky [22], and Mosely [20]. Only the simplest version of this objective, viz., Hayden's, will be defined here. To satisfy the max-min flow criterion, the smallest session rate in the network must be as large as possible. Subject to this constraint, the second-smallest session rate must be as large as possible, etc. Given a network with its link capacities and a set of sessions with their routes and their maximum possible transmission rates, there is a unique set of session rates that satisfies the max-min conditions. The max-min flow criterion will be taken as the definition of throughput fairness in this thesis. Most of the studies mentioned in this paragraph also develop algorithms for computing session rates that are fair according to the various criteria. Many of these algorithms are meant to be implemented in a distributed manner. (In an interesting twist, Jaffe and Bharath-Kumar [18, 1] argue that power is *not* a suitable objective because it is

neither convex nor decentralizable; algorithms to optimize power would be impractical.)

Beyond the issues of defining and computing fair rates is the problem of enforcing them. Several methods have been suggested in the literature. Golestaani and Gallager [12, 8], Thaker and Cain [26], Ibe [14: Section 4.7], Gafni [5: Chapter 5], and Gerla and Staskauskas [11: Section 5.2] use window flow control and adjust the sessions' window sizes to achieve the desired rates. Bially, Gold and Seneff [2], Hayden [13: Chapter 5], Mosely [20], Ibe [14: Section 4.2], Gafni [5: Chapter 3], and Gafni and Bertsekas [6] consider a session input control that produces packet lengths proportional to the session's assigned rate. The time between packet admissions is constant. This model is particularly meaningful for packetized voice traffic: it represents the output of a variable rate vocoder [2]. Oshinsky [22] takes the opposite approach, called *metering.* Time is divided into control intervals whose length is inversely proportional to a session's target rate. The session is permitted to inject some fixed quantity of data into the network during one control interval. A fourth approach, taken by Mukherji [21] and Sauve, Wong and Field [24, 25] is to schedule the use of the links among the various sessions. These studies assume that window flow control is also used, but it is primarily the schedule parameters rather than the window sizes that are adjusted to achieve the desired session rates.

This thesis studies the following new method proposed by Gallager [7, 9] for max-min fair flow control. Each link offers its packet transmission slots to its users by polling them in round robin order. If a session is offered a chance to use a link slot but has no packets ready, then that same slot is offered to the next session, and perhaps the next, etc., until a ready session is found. In each pass of a link's round robin, a session may transmit only one packet. The round robin schedulers for the various links are uncoordinated. In order to prevent excessive queues at the network nodes, window flow control is also employed. The principal contribution of this thesis is a proof that round robin scheduling with windows can indeed be used to achieve max-min throughput fairness.

The main advantage of the round robin method is its simplicity. The desired rates are never explicitly computed, as they are for other fair flow control schemes. The only overhead communication is that required for the window acknowledgments. The window sizes need not be adjusted as network conditions change. An obvious price paid for this simplicity is a lack of flexibility. The strategy is coupled to the max-min criterion and probably cannot be adapted to fairness objectives substantially different from max-min. (Session priorities *can* be implemented, however, by allowing some sessions to transmit more than one packet over a link in each polling cycle.) Another drawback is that large windows are needed to ensure throughput fairness for some networks, and large windows permit large cross-network delays.

However, the thesis also proves certain throughput guarantees that apply even for sessions using small windows.

1.2 Thesis Overview

The system model assumed throughout this thesis is described in Chapter 2. The network is composed of nodes with ample storage connected by point-to-point, error-free links with negligible propagation delays and equal capacities. Uniform packet lengths are also assumed, so that the time required to transmit one packet over one link is the same for all packets and all links. This fundamental time unit is called a *slot*. The network supports a fixed set of S sessions. Virtual circuit routing is used, with no more than H links in the path of any session and no more than N sessions sharing any single link. Window flow control operates on a link-by-link basis, with window sizes of at most W packets. Several link scheduling disciplines are modeled, including round robin, first-come-first-served, and a generalization of these called bounded delay scheduling. A link scheduling discipline is said to have bounded delay if any session with an open window is guaranteed at least a certain service frequency. A variety of deterministic and random models are considered for packet arrivals, but packets are assumed to depart as soon as they reach their destination nodes.

Chapter 4 studies the session throughputs in systems using round robin link scheduling and large windows. There are two major results, one assuming

bursty packet arrivals and one for smooth demand. Theorem 4 asserts that, for Bernoulli packet arrivals, the session throughput rates approach the max-min fair rates as the window sizes increase (provided they all increase at the same rate). If each session has evenly spaced packet arrivals or has such heavy demand that packets are always waiting to enter the network, then the traffic admitted to the network will be rather smooth. For this model, Corollary 1 claims that the long-term average throughputs equal the fair rates if the windows are at least a certain size k_1. Chapter 4 includes several other results for this smooth demand model. In Theorem 2, the throughput of a session during any finite interval is shown to be within a constant k_2 of the max-min fair number of packets (regardless of the length of the interval). This constant k_2 is proportional to the window size, because the system cannot reach a steady state until the buffers upstream of a session's most congested link fill up with packets and the downstream buffers drain. According to Theorem 3, a steady state is eventually reached; thereafter, the unfairness of a session's throughput over any interval is less than another constant $k_3 < k_2$. This steady-state unfairness bound k_3 is less than the minimum acceptable window size k_1, but is independent of the actual window sizes. This suggests that using windows larger than the minimum required value k_1 may cause a longer transient, but probably does not affect the throughput performance of the system in steady state. Corollaries 3 and 4 show that, in steady state, buffers upstream of a session's most congested link are never empty, while

buffers downstream are never full, so that the session accepts every chance offered to it by the round robin scheduler at its most congested link. (Of course, the term "most congested link" must be precisely defined, and if a session has more than one such link, the results are more complicated.) Unfortunately, the minimum window size k_1 for which Corollary 1 proves perfect fairness of the long-term average throughputs is absurdly large and grows exponentially in S. Moreover, recall that for the bursty demand result, no finite window size was sufficient. One wonders how large the windows must be to achieve at least approximate fairness in practice. In Example 3, the session throughput rates are quite unfair unless very large windows are used. The example, however, is somewhat contrived, leaving open the question of performance for "typical" systems. Also included are Examples 6 and 7, which use first-come-first-served link scheduling instead of round robin scheduling. In these examples, the session throughput rates are unfair even if large windows are used. The throughput rates are shown to be very sensitive to the relative window sizes of competing sessions and to the initial conditions of the system, even if the windows are large. This contrasts sharply with the round robin results.

Chapter 4 seeks max-min fair throughputs, asking how large the windows must be. Chapter 5 takes a different approach. It assumes only that the window sizes are at least two packets and asks how unfair the throughputs can be. Bounded-delay link scheduling is assumed, and bursty demand is

permitted. The chapter presents two theorems that differ in their assumptions about how many packets can be stored at a session's source. Theorem 6 shows that if round robin scheduling is used, then the throughput rate of a session with demand rate λ packets/slot and unlimited source buffering is at least $\min[\lambda, 1/N]$ packets/slot. This throughput guarantee is within a factor of N of the max-min fair rate. The analogous bound for first-come-first-served scheduling is roughly $\min[\lambda, 1/(N \cdot W)]$ packets/slot. This throughput guarantee is worse than the round robin guarantee by roughly a factor of W if λ is large. Theorem 5 gives other throughput guarantees for sessions whose source buffers are finite but hold at least two packets. These guarantees are too complicated to describe here, but if the session's demand rate $\lambda \geq 1$ packet/slot, then the guarantees match those given above for unlimited source buffering. Round robin Example 1 (Section 3.2) and first-come-first-served Example 2 (Section 3.3) show a session x with demand rate $\lambda = 1$ packet/slot whose throughput rate (for either source buffering assumption) matches the theoretical lower bound. † Example 3 (Section 4.4.2) shows that, with round robin scheduling, the throughput rate of a session with heavy demand and

† In the first-come-first-served example, however, session x must have at least one window of size two packets, while the other sessions may have arbitrarily large windows. I suspect that if all sessions are required to have the same window size W and W is arbitrary (except that $W \geq 2$ packets), then the worst case throughput for first-come-first-served systems is roughly half that of round robin systems. Moreover, a different implementation of first-come-first-served scheduling than that assumed here might perform better.

small windows really can be unfair by a factor proportional to N. For first-come-first-served scheduling, Example 7 (Section 4.6.2) shows an unfairness factor proportional to N and Example 2 (Section 3.3) shows an unfairness factor roughly equal to W.

While throughput fairness is the primary focus of this study, cross-network delay is also of interest. Consider a session x whose window size w is at least two packets. Of course, Little's formula [19, 4] can take a given lower bound of R packets/slot for x's throughput rate (such as the bounds given in Chapters 4 and 5) and generate an upper bound of roughly $w \cdot H/R$ slots for the *average* cross-network delay of x's packets. Theorem 1 of Chapter 3, however, derives an upper bound on delay that applies to *each* packet and is tighter in some cases than the bound from Little's formula. The theorem assumes bounded-delay link scheduling. For round robin scheduling, the upper bound on x's cross-network delay is roughly $w \cdot H \cdot N$ slots. The analogous bound for first-come-first-served scheduling is roughly $w \cdot H \cdot N \cdot W$ slots -- worse by a factor of W. Round robin Example 1 and first-come-first-served Example 2 show packets whose cross-network delays come close to the theoretical upper bounds. †

† In the first-come-first-served example, however, session x must have at least one window of size two packets, while the other sessions may have arbitrarily large windows. I suspect that if all sessions are required to have the same window size W and W is arbitrary (except that $W \geq 2$ packets), then the worst case delay for first-come-first-served systems is roughly twice that of round robin systems. Moreover, a different implementation of first-come-first-served scheduling than that assumed here might perform better.

Chapter 6 discusses the practical implications of this research, more thoroughly compares the round robin method with the fair flow control strategies mentioned in Section 1.1, and offers suggestions for future study. A glossary of notation appears at the end of the document.

2. SYSTEM MODEL

This chapter presents a system model to be studied in Chapters 3, 4 and 5. The model features uniform link capacities, uniform packet lengths, a fixed set of users, virtual circuit routing, and link-by-link window flow control. Several link scheduling disciplines are modeled, including round robin, first-come-first-served, and a generalization of these called "bounded delay" scheduling. A variety of deterministic and random models are considered for packet arrivals, but packets are assumed to depart as soon as they reach their destination nodes.

2.1 Nodes, Links, Packets, and Time

The network consists of store-and-forward nodes joined by point-to-point links. A link allows communication in only one direction. If two nodes are connected by link(s) in one direction, then they must be connected by at least one link in the reverse direction so that flow control acknowledgments can be returned. Except for this restriction, two nodes may have any number of links connecting them.

Links are perfectly reliable, i.e., they never lose or corrupt data, and they never fail. Nodes, too, are perfectly reliable, and the storage capacity of each node is infinite.

Data are transmitted through the network in packets of equal length. A

time interval during which one packet is transmitted over a link is called a *time slot* for that link. All links have the same capacity; hence all time slots have the same length. A packet experiences no processing delay at a node, other than a possible queuing delay as it waits for transmission. A packet experiences no propagation delay on a link. The transmission time slots of all links are synchronized; hence the entire network operates with slotted time. Although the operation of the system *during* a time slot will occasionally be discussed, a discrete-time system model will normally be used in which the t^{th} discrete-time instant, called *time* t, refers to the *end* of the t^{th} time slot. The model begins at time 0.

It is often necessary to refer to intervals longer than one slot. Let $[s, t]$ denote the interval from the beginning of slot s to the end of slot t; if $s > t$, then $[s, t]$ is null. Define $(s, t]$, $[s, t)$, and (s, t) as follows:

(1) $$(s, t] = [s+1, t]$$

(2) $$[s, t) = [s, t-1]$$

(3) $$(s, t) = [s+1, t-1]$$

2.2 Sessions, Routes, Throughputs, and Delays

The network supports one-way communication activities called *sessions*. Each session is assigned a path (i.e., a sequence of appropriately directed links) through the network from its *origin* node to its *destination* node; data packets for the session are transmitted along this path. Several sessions may have the

same origin and destination nodes. The set of sessions using the network is fixed. Let S denote the number of sessions using the network, and let $N'(l)$ denote the number of sessions using link l.

While all links have global identifiers, it is often convenient to index links in the order in which some session uses them. Therefore, let $H(x)$ denote the number of links in the path of session x, and let H be the maximum number of links in the path of any session:

$$H = \max_x H(x) \tag{4}$$

For each session x and for $h = 1, 2, \ldots, H(x)$, define *hop* h as the h^{th} link in the path of x, including the related functions in the node at the input end of that link. To streamline the analysis, packet arrivals and departures are modeled as transmissions over dummy hops. The session's source is *hop* 0. The session's sink is *hop* $H(x)+1$. The source and sink are considered to be hops but not links. For $h = 1, 2, \ldots, H(x)$, let $N(x, h)$ denote the number of sessions using hop h of session x, including x itself. Let $N(x)$ be the maximum number of sessions using any link in the path of x :

$$N(x) = \max_{1 \le h \le H(x)} N(x, h) \tag{5}$$

Let N be the maximum number of sessions sharing any link in the network:

$$N = \max_x N(x) = \max_l N'(l) \tag{6}$$

Packets waiting to be transmitted over hop h of session x, $0 \leq h \leq H(x)+1$, are said to be in *buffer h*. The number of packets in buffer h of session x at time t is called the *buffer level* $B(x, h, t)$. For convenience, packet arrivals for session x are modeled as occasional services at hop 0. Buffer 0 is assumed to contain an infinite number of packets at time 0. In each time slot, the session's source (hop 0) transfers either one packet or no packets from buffer 0 to buffer 1. Therefore, for all times $t \geq 0$,

$$B(x, 0, t) = \infty \tag{7}$$

The only significance of (7) is that buffer 0 is never empty. The initial levels $B(x, h, 0)$ of the buffers $h = 1, 2, \ldots, H(x)$ are assumed to be finite but are not necessarily zero.† It is assumed that the initial level $B(x, H(x)+1, 0)$ of buffer $H(x)+1$ is at most one, and that the session's sink is very fast. ‡ That is, whenever $B(x, H(x)+1, t-1) > 0$, hop $H(x)+1$ removes one packet from buffer $H(x)+1$ during slot t. Therefore, for all times $t \geq 0$,

---

† Since the initial buffer levels can be positive, the assumption of a fixed set of sessions is less restrictive than it appears. The approach here is equivalent to studying a more realistic model, one with a dynamic set of sessions, during an interval in which no existing sessions are terminated and no new sessions are initiated.

‡ Fast session sinks have been assumed for simplicity. This assumption could be relaxed by using a sink model similar to the source model. Straightforward modifications would adapt Chapter 4 to such a model. It is not clear whether Chapters 3 and 5 could also be generalized.

(8) $$B(x, H(x)+1, t) \leq 1$$

The throughput (measured in packets) of session x over hop h, $0 \leq h \leq H(x)+1$, during interval $(s, t]$ is denoted by $P(x, h, s, t)$. For $s \geq t$, $P(x, h, s, t)$ is defined to be zero. Note that, for $s \leq t \leq u$,

(9) $$P(x, h, s, u) = P(x, h, s, t) + P(x, h, t, u)$$

If link l is hop h for session x, then $P'(x, l, s, t)$ is defined to equal $P(x, h, s, t)$. There is a simple and obvious relationship between buffer levels and throughputs. For any session x, any buffer h of x in the range $1 \leq h \leq H(x)+1$, and any times s and t such that $0 \leq s \leq t$,

(10) $$B(x, h, t) = B(x, h, s) + P(x, h-1, s, t) - P(x, h, s, t)$$

The long-term average throughput $R_A(x)$ of a session x is defined as follows:

(11) $$R_A(x) = \lim_{t \to \infty} \frac{P(x, H(x), 0, t)}{t}$$

This limit may not exist for systems with irregular demand.

Sequence numbers are assigned to the packets of each session x. If buffers $h = 1, 2, \ldots, H(x)$ contain any packets at time 0, then the one farthest downstream is called packet 1; if these buffers are initially empty, then the first packet to arrive at buffer 1 after time 0 is called packet 1. The packet following packet 1 is called packet 2, etc. (If buffer $H(x)+1$ initially contains a packet, that packet gets no sequence number.)

Let $\Upsilon(x, h, p)$ denote the time slot during which packet $p \geq 1$ of session x crosses hop h, $0 \leq h \leq H(x)+1$. For convenience, let $\Upsilon(x, h, p)$ be defined for all integers p, with $\Upsilon(x, h, p) = 0$ if $p < 1$ or if packet p is farther downstream than buffer h at time 0. If a packet $p \geq 1$ gets stuck in a buffer $\bar{h}$, $0 \leq \bar{h} \leq H(x)$, and never advances, take $\Upsilon(x, h, p) = \infty$ for $\bar{h} \leq h \leq H(x)+1$. Consider the following claim:

(12) $\Upsilon(x, H(x)+1, p) \leq \Upsilon(x, H(x), p+1)$ for all integers p

For $p < 1$, (12) holds because its left-hand side equals zero. For $p \geq 1$, (12) holds because the assumptions about buffer $H(x)+1$ and hop $H(x)+1$ imply that

$$\Upsilon(x, H(x)+1, p) = \Upsilon(x, H(x), p) + 1 \leq \Upsilon(x, H(x), p+1)$$

For each packet $p \geq 1$ of session x that gets beyond buffer 1 (i.e., for which $\Upsilon(x, 1, p) < \infty$), define the *cross-network delay* $\Xi(x, p)$ as follows:

(13) $$\Xi(x, p) = \Upsilon(x, H(x), p) - \Upsilon(x, 1, p) + 1$$

This measures the transmission delays across hops 1 through $H(x)$, inclusive, plus the queuing delays in buffers 2 through $H(x)$, inclusive.

2.3 Link-by-Link Window Flow Control

The capacity (in packets) of a buffer h of a session x, $0 \leq h \leq H(x)+1$, is called the *window size* associated with that buffer and is denoted by

$W(x, h)$. For each session x, buffer 0 has infinite capacity:

$$W(x,0) = \infty \tag{14}$$

The capacity $W(x, 1)$ of buffer 1 may be either finite or infinite; this permits a wider variety of packet arrival models. The buffers h in the range $2 \leq h \leq H(x)$ must have finite capacity. The capacity of buffer $H(x)+1$ is assumed to be at least two but finite:

$$2 \leq W(x, H(x)+1) < \infty \tag{15}$$

It follows from (15) and (8) that

$$B(x, H(x)+1, t) < W(x, H(x)+1) \tag{16}$$

for all times $t \geq 0$. In other words, buffer $H(x)+1$ is never full. Buffers 2 through $H(x)+1$ are required to be finite for two reasons -- to bound the cross-network delay, and to keep individual sessions from consuming grossly unfair amounts of link capacity. Each result in Chapters 3, 4 and 5 requires its own additional assumptions about the window sizes. The maximum window size for any session x at any buffer h in the range $1 \leq h \leq H(x)$ is denoted by W' :

$$W' = \max_{x,h:\ 1 \leq h \leq H(x)} W(x, h) \tag{17}$$

The maximum window size for any session x at any buffer h in the range $2 \leq h \leq H(x)+1$ is denoted by W'' :

$$W'' = \max_{x,h:\, 2 \le h \le H(x)+1} W(x, h) \tag{18}$$

To keep each buffer h in the range $1 \le h \le H(x)+1$ from overflowing, the following restriction is placed on the flow over hop $h-1$ during every slot $t \ge 1$: if buffer h is full at the end of slot $t-1$, the window is said to be *closed*, and session x may not transmit a packet over hop $h-1$ during slot t. In other words,

$$B(x, h, t-1) = W(x, h) \quad \text{implies} \quad P(x, h-1, t-1, t) = 0 \tag{19}$$

This restriction is implemented using logical quantities called *permits*. Buffer h has $W(x, h)$ permits permanently associated with it. Every packet waiting in buffer h holds a permit for buffer h, and any leftover permits for buffer h are stored at hop $h-1$. A packet being transmitted over hop $h-1$ must carry with it a permit for buffer h. (If none are available at hop $h-1$, then the packet may not be transmitted.) Whenever a packet is transmitted over hop h (i.e., removed from buffer h) during a time slot t, the packet relinquishes its permit for buffer h, and hop h returns this permit upstream to hop $h-1$ during that same slot t.† The return of permits is accomplished in the

† In this model, link transmission is error-free. Hence, it is possible for hop h to know, at the beginning of a slot t, that a packet p will be successfully transmitted over hop h during slot t. Therefore, hop h can safely send packet p's permit for buffer h back to hop $h-1$ during slot t. In a real network, with imperfect transmission, the error control process (i.e., link layer protocol) for hop h could have its own storage area where the packet p is transferred during the slot t in which its transmission is first attempted. With this arrangement, hop h could still send packet p's permit back to hop $h-1$ during slot t without risking buffer overflow.

following way. A permit for buffer 1 is returned by having the session's origin node notify the session's source (i.e., hop 0). A permit for a buffer h in the range $2 \leq h \leq H(x)+1$ is returned to hop $h-1$ by transmitting a notice over some link with direction opposite to hop $h-1$. This notice requires few bits and can be piggybacked onto a data packet if any are available. Therefore the link capacity required to implement permits will be ignored. †

2.4 Link Scheduling

At any link, packets for any particular session are transmitted in the order in which they arrived from their preceding hop. Packets from different sessions, however, are not necessarily transmitted in order of arrival. Each link has a scheduler to decide which session will use the link during each time slot. Various scheduling disciplines are possible. This section describes round robin scheduling, first-come-first-served scheduling, and a generalization of these disciplines called *bounded delay* scheduling.

---

† Consider a session x that is the sole user of each link in its path. Suppose x has heavy demand; i.e., it inserts a packet into buffer 1 whenever that buffer is not full. If x's window sizes are at least two, then its long-term average throughput $R_A(x)$ equals one, the link capacity. However, if $W(x, h) = 1$ for some h, then $R_A(x)$ can be no more than ½. For this reason, a window size of two seems to be the smallest practical value.

2.4.1 Round Robin Scheduling

To implement a round robin discipline, the scheduler for link l consults a fixed data structure consisting of session identifiers arranged in a directed ring. Each session using l appears exactly once on this *round robin ring.* The link scheduler also maintains a variable called the *ring position* that points to some session on the ring. Whenever a session x transmits a packet over l in a time slot, the ring position is updated to x during that slot.

Consider the scheduling of slot t at link l. Let session x be the ring position at the end of slot $t-1$. Let y be the session immediately following x on the ring. During slot t, the link scheduler searches the ring, starting with y, until it finds the first session z that has both packet(s) and permit(s) available; i.e., z must satisfy

$$B(z, h, t-1) > 0$$

and

$$B(z, h+1, t-1) < W(z, h+1)$$

where h is the hop number of link l for z. A packet for session z is transmitted over l and the ring position is updated to z during slot t. If the ring is searched through session x without success, then the search stops after x, the ring position remains at x, and nothing is transmitted over l during slot t.

Each session examined in the search described above is said to have been offered one *chance* to use link l. Let $C'(x, l, s, t)$ denote the number of

chances offered to session x at link l during interval $(s, t]$. If h is the hop number of link l for session x, then $C(x, h, s, t)$ is defined to equal $C'(x, l, s, t)$. For $s \geq t$, define $C'(x, l, s, t)$ and $C(x, h, s, t)$ to be zero.

2.4.2 First-Come-First-Served Scheduling

First-come-first-served link scheduling is complicated by the window flow control mechanism. With this discipline, a packet waiting in buffer h of session x, $1 \leq h \leq H(x)$, seizes a permit for its next buffer $h+1$ as early as possible. Note that the packet may enter buffer h before or after the permit needed for buffer $h+1$ arrives. Once both the packet and its permit for buffer $h+1$ have arrived at hop h, the packet is said to be *authorized* for transmission over hop h, and a future transmission slot on that link is reserved for that session.

First-come-first-served scheduling transmits packets over a link in order of their authorization times. To this end, the scheduler for each link l maintains a first-in-first-out *transmitter queue*. If a packet for session x becomes authorized to use l during slot t, then a reservation for x is added to the tail of the transmitter queue during slot t. Associated with link l is a fixed *tie-breaking list*; each session using l appears exactly once in this list. If packets for several sessions become authorized to use l during slot t, then their reservations are added to the tail of the transmitter queue during slot t in the order in which the sessions appear in l's tie-breaking list. At the beginning of

slot $t+1$, the link scheduler for l notes which session holds the reservation at the head of the transmitter queue. That reservation is removed from the queue and a packet for that session is transmitted over l during slot $t+1$.

Consider a session x that uses link l as its hop h, $1 \leq h \leq H(x)$. Since x can have at most $W(x, h)$ packets and $W(x, h+1)$ permits for buffer $h+1$ waiting at hop h, the number of reservations for x in l's transmitter queue (i.e., the number of authorized packets) can be no more than $\min[W(x, h), W(x,h+1)]$; note that †

$$\min\,[\,W(x, h), W(x, h+1)\,] \leq W(x, h+1) \leq W'' < \infty \tag{20}$$

2.4.3 Bounded Delay Scheduling

This section describes a family of link scheduling disciplines called *bounded delay disciplines.* Consider a link l used by at least one session. The scheduling discipline for l is said to have *bounded delay* if there exists a positive integer $A'(l)$ (called a *schedule delay bound*) such that, for all sessions x using l and for all packets $p \geq 1$ of x,

---

† Window size $W(x, H(x)+1)$ was required to be finite so that the number of packets for session x in the transmitter queue at hop $H(x)$ would be bounded even in the case where $H(x) = 1$ and $W(x, H(x)) = W(x, 1) = \infty$. It will be shown in Section 2.4.3 that bounded transmitter queues make first-come-first-served a bounded delay discipline; such disciplines offer delay and throughput guarantees to be studied in Chapters 3 and 5.

(21)

$$\Upsilon(x, h, p)$$

$$\leq \max [\, \Upsilon(x, h-1, p), \Upsilon(x, h, p-1), \Upsilon(x, h+1, p - W(x, h+1)) \,] + A'(l)$$

where h is the hop number of link l for session x, $1 \leq h \leq H(x)$. In other words, once packet p is in buffer h, and there are no packets older than p in buffer h, and there is room in buffer $h+1$, packet p is guaranteed to be transmitted over hop h within $A'(l)$ time slots. Apart from satisfying (21), the scheduling decision for a slot t at link l may be any deterministic function of time t itself, of the initial levels $B(x, h, 0)$ of all buffers h of all sessions x, $0 \leq h \leq H(x)+1$, and of the throughputs $P(x, h, \tau-1, \tau)$ of all sessions x over all hops h, $0 \leq h \leq H(x)+1$, at all past times τ, $1 \leq \tau < t$.

Note that round robin is a bounded delay discipline, with $A'(l) = N'(l)$. To see that first-come-first-served also has bounded delay, consider a session x that uses link l as its hop h. Consider a packet $p \geq 1$ of x waiting in buffer h. Suppose there are no packets for x in buffer h that are older than p, and suppose the window for buffer $h+1$ of session x is open. Consequently, p is authorized for transmission over hop h. Consider how many packets could be ahead of p in link l's transmitter queue. It was given that there are no packets for session x ahead of p. There can be at most $N'(l)-1$ sessions other than x using l. It was explained in Section 2.4.2 that a session can have no more than W'' packets in any link transmitter queue. Therefore, the number of packets ahead of p in l's transmitter queue is no more than $[N'(l)-1] \cdot W''$. In

addition to the packets ahead of it, packet p itself must be transmitted within the $A'(l)$ time slots. Hence, first-come-first-served is a bounded delay discipline, with $A'(l) = [N'(l)-1]\cdot W'' + 1$. An example of a link scheduling discipline *without* bounded delay is a priority scheme where the session with highest priority could monopolize a link for an indefinite period of time.

In a system where each link l has a bounded delay scheduler with delay bound $A'(l)$, define $A(x, h)$ to equal $A'(l)$ if link l is hop h for session x. Also define $A(x)$ to be the largest schedule delay bound at any link in the path of session x :

$$A(x) = \max_{1 \le h \le H(x)} A(x, h) \tag{22}$$

2.5 Demand

Recall that packet arrivals for a session x are modeled as occasional services at hop 0. This section explains the session demand model in more detail. During random time slots $t \ge 1$, the session source (i.e., hop 0) attempts to place one packet (taken from the infinite supply in buffer 0) into buffer 1. If the window for buffer 1 is closed, i.e., if $B(x, 1, t-1) = W(x, 1)$, then the attempt fails and the packet transfer does not take place. The number of such attempts (called *chances at hop* 0) during interval $(s, t]$ is denoted by $C(x, 0, s, t)$. For $s \ge t$, $C(x, 0, s, t)$ is defined to be

zero. † The long-term average number of chances per time slot for session x at hop 0 (if this average exists) is called the *demand rate* $\lambda(x)$. The results of Chapters 3 and 5 do not require the existence of this average for every session. Note that $\lambda(x) \leq 1$. ‡

The sample space of the demand model is denoted by Ω. A single sample point ω in Ω determines an entire sample path of demand for the whole network; i.e., ω determines $C(x, 0, t-1, t)$ for all sessions x and all times $t \geq 1$. Since the demand is the only random element in the system model, a single sample point ω in Ω also determines the evolution of the entire system after time 0. The σ-algebra for the demand model is the one generated by events of the following type: $C(x, 0, t-1, t)$ is specified for a single session x and a single time $t \geq 1$, while the demand for other sessions and other times is arbitrary.

Different sections in Chapters 3, 4 and 5 make different assumptions about the probability measure of the demand model. Some theorems require the demand of each session to be extremely regular, almost deterministic. Other

---

† Although $C(x, h, s, t)$ is defined for $h > 0$ only in the context of round robin link scheduling, $C(x, 0, s, t)$ is defined here regardless of the link scheduling discipline.

‡ A session x whose actual demand rate is greater than one packet per slot can be modeled with $\lambda(x) = 1$, since the network cannot offer x a throughput rate greater than one packet per slot even if x is the sole user of each link in its path.

results permit the demand process $C(x, 0, t-1, t)$ of a session x to be Bernoulli. One section assumes only that the times between chances for a session at hop 0 are independent and identically distributed. Some results require that the demand processes of the various sessions be independent; others permit dependence. One theorem makes no demand assumptions at all. The one set of demand assumptions under which all the results of Chapters 3, 4 and 5 hold is the heavy demand assumption, viz., that $C(x, 0, t-1, t) = 1$ for all sessions x and all times $t \geq 1$.

2.6 System Specification

A system is fully specified by describing the following items: the network topology, the set of sessions using the network, the sessions' paths, the initial buffer levels, the window sizes, the scheduling discipline (e.g., round robin or first-come-first-served), the schedule parameters (e.g., the rings for round robin scheduling or the tie-breaking lists for first-come-first-served scheduling), the initial schedule state (e.g., the initial ring positions for round robin scheduling or the initial transmitter queues for first-come-first-served scheduling), and the probability measure of the demand model.

2.7 Miscellaneous Bounds

Some bounds on quantities defined in this chapter are listed below for easy reference.

$1 \leq H(x) \leq H < \infty$	for all x
$1 \leq N'(l) = N(x,h) \leq N(x) \leq N \leq S < \infty$	for all l, x, h such that link l is hop h for session x
$1 \leq A'(l) = A(x,h) \leq A(x) < \infty$	for all l, x, h such that link l is hop h for session x
$W(x,0) = \infty$	for all x
$1 \leq W(x,1) \leq \infty$	for all x
$1 \leq W(x,h) < \infty$	for all x, for $2 \leq h \leq H(x)$
$2 \leq W(x,H(x)+1) < \infty$	for all x
$W(x,h) \leq W'$	for all x, for $1 \leq h \leq H(x)$
$1 \leq W' \leq \infty$	
$W(x,h) \leq W''$	for all x, for $2 \leq h \leq H(x)+1$
$2 \leq W'' < \infty$	
$0 \leq B(x,h,t) \leq W(x,h)$	for all x, for $0 \leq h \leq H(x)$, for $t \geq 0$
$0 \leq B(x,H(x)+1,t) \leq 1 < W(x,H(x)+1)$	for all x, for $t \geq 0$

$B(x, 0, t) = \infty$	for all x, for $t \geq 0$
$B(x, h, t) < \infty$	for all x, for $1 \leq h \leq H(x)+1$, for $t \geq 0$
$0 \leq P(x, h, s, t) = P'(x, l, s, t) \leq t - s$	for all x, h, l such that link l is hop h for session x, for $t \geq s \geq 0$
$0 \leq P(x, 0, s, t) \leq t - s$	for all x, for $t \geq s \geq 0$
$0 \leq P(x, H(x)+1, s, t) \leq t - s$	for all x, for $t \geq s \geq 0$
$0 \leq C(x, h, s, t) = C'(x, l, s, t) \leq t - s$	for all x, h, l such that link l is hop h for session x, for $t \geq s \geq 0$
$0 \leq C(x, 0, s, t) \leq t - s$	for all x, for $t \geq s \geq 0$
$0 \leq R_A(x) \leq 1$	for all x whose avg. throughput exists
$0 \leq \lambda(x) \leq 1$	for all x whose avg. demand exists
$0 \leq \Upsilon(x, h, p) \leq \infty$	for all x, for $0 \leq h \leq H(x)+1$, for all integers p
$1 \leq \Xi(x, p) \leq \infty$	for all x, for all $p \geq 1$ such that $\Upsilon(x, 1, p) < \infty$

3. PACKET DELAY

This chapter studies the cross-network delay of packets for a particular session x in a system with bounded-delay link scheduling. The window sizes $W(x, h)$ for buffers h in the range $2 \leq h \leq H(x)+1$ are assumed to be at least two but finite. The capacity $W(x, 1)$ of buffer 1 is arbitrary, possibly even infinite. The window sizes of the other sessions in the network are arbitrary. The demands of the sessions, including x, are arbitrary; session demand rates need not exist. Theorem 1 shows that the cross-network delay for each packet of x is at most $\left[\sum_{h=2}^{H(x)} W(x, h)\right] \cdot A(x) + 1$. †

It was explained in Section 2.4.3 that round robin scheduling and first-come-first-served scheduling are bounded delay disciplines, with schedule delay bounds $A'(l)$ of $N'(l)$ and $N'(l) \cdot W'' - W'' + 1$, respectively. Therefore, the cross-network delay bounds of Theorem 1 for round robin systems and first-come-first-served systems are $\left[\sum_{h=2}^{H(x)} W(x, h)\right] \cdot N(x) + 1$ and

---

† If the average throughput and the average cross-network delay per packet exist for session x, then Little's formula [19, 4] gives the following upper bound on the *average* cross-network delay per packet: $\left[1 + \sum_{h=2}^{H(x)} W(x, h)\right] \cdot \frac{1}{R_A(x)}$. This bound may be tighter than the bound of Theorem 1. Note, however, that Theorem 1's bound applies to *each* packet of session x.

$\left[\sum_{h=2}^{H(x)} W(x, h)\right] \cdot [N(x) \cdot W'' - W'' + 1] + 1$, respectively. Example 1 shows a round robin system where one packet of session x has a cross-network delay that matches the bound of Theorem 1. Example 2 shows a first-come-first-served system where one packet of session x has a cross-network delay of $\left[\sum_{h=2}^{H(x)} W(x, h) - 1\right] \cdot [N(x) \cdot W'' - W'' + 1] + 1$, which is close to the bound of Theorem 1. † Clearly, the delay guarantees afforded by this theorem for round robin systems are superior to those for first-come-first-served systems. (It is *not* being claimed that round robin scheduling *always* offers lower packet delays or fairer packet delays than first-come-first-served scheduling.)

† In Example 2, it is critical that $W(x, H(x)) = 2$.

3.1 Theorem 1: Bound on Packet Delay

Suppose a system has been specified that satisfies the assumptions of Chapter 2 and uses a bounded delay scheduling discipline. Let x be some session. The capacity $W(x, 1)$ of buffer 1 of x is arbitrary -- it may even be infinite. Suppose that

$$2 \leq W(x, h) < \infty \quad \text{for } 2 \leq h \leq H(x)+1 \tag{23}$$

The window sizes of the other sessions in the network are arbitrary (i.e., these window sizes only need to satisfy the basic assumptions of Chapter 2). The demands of the sessions, including session x, are arbitrary. It follows that, for each packet $p \geq 1$ of session x such that $\Upsilon(x, 0, p) < \infty$,

$$\Upsilon(x, h, p) < \infty \quad \text{for } 1 \leq h \leq H(x)+1 \tag{24}$$

$$\Xi(x, p) \leq \left[\sum_{h=2}^{H(x)} W(x, h)\right] \cdot A(x) + 1 \tag{25}$$

In other words, every packet of x that enters buffer 1 eventually leaves the network and has bounded cross-network delay.

Proof of Theorem 1

Let us clarify the scheduling assumptions. It follows from (21) that, for all packets $p \geq 1$ of x and all hops h in the range $1 \leq h \leq H(x)$,

$$\Upsilon(x, h, p)$$

$$\leq \max [\, \Upsilon(x, h-1, p), \Upsilon(x, h, p-1), \Upsilon(x, h+1, p-W(x, h+1))\,] + A(x, h)$$

(26)
$$\leq \max [\, \Upsilon(x, h-1, p), \Upsilon(x, h, p-1), \Upsilon(x, h+1, p-W(x, h+1))\,] + A(x)$$

Also recall this assumption about the operation of hop $H(x)+1$:

(27) $$\Upsilon(x, H(x)+1, p) = \Upsilon(x, H(x), p) + 1 \quad \text{for } p \geq 1$$

Now it will be proved by contradiction that (24) holds for each packet $p \geq 1$ of session x such that $\Upsilon(x, 0, p) < \infty$. If this were not the case, then there would be some smallest positive integer $\bar{p}$ such that

(28) $$\Upsilon(x, 0, \bar{p}) < \infty$$

and

(29) $$\Upsilon(x, h, \bar{p}) = \infty \quad \text{for some } h,\ 1 \leq h \leq H(x)+1$$

Since $\bar{p}$ is the smallest such value,

(30) $$\Upsilon(x, h, p) < \infty \quad \text{for } 0 \leq h \leq H(x)+1, \text{ all integers } p < \bar{p}$$

Let $\bar{h}$ be the smallest integer in the range $1 \leq \bar{h} \leq H(x)+1$ for which (29) holds; i.e.,

(31) $$\Upsilon(x, \bar{h}, \bar{p}) = \infty$$

and

(32) $$\Upsilon(x, h, \bar{p}) < \infty \quad \text{for } 1 \leq h < \bar{h}$$

Now it will be shown that

$$\Upsilon(x, \bar{h}, \bar{p}) < \infty \tag{33}$$

There are three cases to consider. If $\bar{h} = 1$, then (33) follows from (26), (28) and (30). If $2 \leq \bar{h} \leq H(x)$, then (33) follows from (26), (32) and (30). If $\bar{h} = H(x)+1$, then (33) follows from (27) and (32). Hence (33) is proved. Note that (33) contradicts (31). This completes the proof of (24).

Next (25) will be proved. Let $p' \geq 1$ be any packet of session x such that $\Upsilon(x, 0, p') < \infty$; p' will be fixed for the remainder of this section. By (24), $\Upsilon(x, 1, p') < \infty$, so the cross-network delay of packet p' is well-defined. It must be shown that

$$\Xi(x, p') \leq \left[\sum_{h=2}^{H(x)} W(x, h)\right] \cdot A(x) + 1 \tag{34}$$

If $H(x) = 1$, then (34) follows immediately from definition (13); so assume that

$$H(x) \geq 2 \tag{35}$$

First, let us justify the following claim for integers h and p.

$$\Upsilon(x, h, p) \leq \Upsilon(x, 1, p') \quad \text{for } 1 \leq h \leq H(x),\ p \leq \left[p' - \sum_{\underline{h}=2}^{h} W(x, \underline{h})\right] \tag{36}$$

Inequality (36) follows from the buffer capacity constraints:

$$\begin{aligned}
\Upsilon(x,\ 1,\ p') &\geq \Upsilon(x,\ 2,\ p' - W(x,2)) \\
&\geq \Upsilon(x,\ 3,\ p' - W(x,2) - W(x,3)) \\
&\ \ \vdots \\
&\geq \Upsilon(x,\ h,\ p' - \sum_{\underline{h}=2}^{h} W(x,\underline{h})) \\
&\geq \Upsilon(x,\ h,\ p)
\end{aligned}$$

Note that (36) holds if $p < 1$ (in which case $\Upsilon(x, h, p) = 0$). Moreover, (36) holds even if packet p' is farther downstream than buffer 1 at time 0 (in which case every element in the chain of inequalities above equals zero).

Next, let us define $\Theta(h, p)$ for each hop h of x in the range $1 \leq h \leq H(x)$ and every integer p :

$$(37) \quad \Theta(h, p) = \Upsilon(x, 1, p') + \max\left[0,\ p - p' + H(x) - h + \sum_{\underline{h}=2}^{h} W(x,\underline{h})\right] \cdot A(x)$$

Note that

$$(38) \quad \Theta(h, p-1) = \Theta(h+1, p - W(x, h+1)) \qquad \text{for } 1 \leq h \leq H(x)-1, \text{ all } p$$

and

$$(39) \quad \Theta(h, p-1) = \Theta(h, p) - A(x) \quad \text{if } \left[p - p' + H(x) - h + \sum_{\underline{h}=2}^{h} W(x,\underline{h})\right] \geq 1$$

Also note that, by (37) and (23),

$$(40) \qquad \Theta(h, p-1) \geq \Theta(h-1, p) \qquad \text{for } 2 \leq h \leq H(x), \text{ all } p$$

Let us prove the following claim:

(41)
$$\Upsilon(x, h, p) \leq \Theta(h, p) \text{ for } 1 \leq h \leq H(x), \left[p'+1-2\cdot\sum_{\underline{h}=2}^{H(x)} W(x, \underline{h})\right] \leq p \leq p'$$

The proof is by induction on p. The base cases, viz.,

$$\left[p'+1-2\cdot\sum_{\underline{h}=2}^{H(x)} W(x, \underline{h})\right] \leq p \leq \left[p'-\sum_{\underline{h}=2}^{H(x)} W(x, \underline{h})\right]$$

are easy to prove: by (36) and definition (37),

$$\Upsilon(x, h, p) \leq \Upsilon(x, 1, p') \leq \Theta(h, p)$$

for these values of p and for $1 \leq h \leq H(x)$. For the induction step, consider an integer $\hat{p}$ in the range

(42)
$$\left[p'+1-\sum_{\underline{h}=2}^{H(x)} W(x, \underline{h})\right] \leq \hat{p} \leq p'$$

The induction hypothesis asserts that

(43)
$$\Upsilon(x, h, p) \leq \Theta(h, p) \text{ for } 1 \leq h \leq H(x), \left[p'+1-2\cdot\sum_{\underline{h}=2}^{H(x)} W(x, \underline{h})\right] \leq p \leq \hat{p}-1$$

It must be shown that

(44)
$$\Upsilon(x, h, \hat{p}) \leq \Theta(h, \hat{p}) \quad \text{for } 1 \leq h \leq H(x)$$

First (44) will be proved for small values of h. Let h' be the largest hop in the range $1 \leq h' \leq H(x)-1$ such that $\left[\sum_{\underline{h}=2}^{h'} W(x, \underline{h})\right] \leq p'-\hat{p}$. By (36) and

definition (37),

$$(45) \qquad \Upsilon(x, h, \hat{p}) \leq \Upsilon(x, 1, p') \leq \Theta(h, \hat{p}) \quad \text{for } 1 \leq h \leq h'$$

For hops h in the range $h' \leq h \leq H(x)$, the proof of (44) will be by induction on h. The base case $h = h'$ is covered by (45) above. For the induction step, consider a hop $\hat{h}$ (if any) in the range

$$(46) \qquad h'+1 \leq \hat{h} \leq H(x)-1$$

(The case $\hat{h} = H(x)$ will be treated separately.) The induction hypothesis asserts that

$$(47) \qquad \Upsilon(x, \hat{h}-1, \hat{p}) \leq \Theta(\hat{h}-1, \hat{p})$$

It must be shown that

$$(48) \qquad \Upsilon(x, \hat{h}, \hat{p}) \leq \Theta(\hat{h}, \hat{p})$$

If $\hat{p} < 1$, the proof of (48) is trivial, since $\Upsilon(x, \hat{h}, \hat{p}) = 0$ in this case. If $\hat{p} \geq 1$, first apply (26) and induction hypotheses (47) (for the induction on h) and (43) (for the induction on p):

$$\Upsilon(x, \hat{h}, \hat{p})$$

$$(49) \qquad \leq \max [\, \Upsilon(x, \hat{h}-1, \hat{p}), \Upsilon(x, \hat{h}, \hat{p}-1), \Upsilon(x, \hat{h}+1, \hat{p}-W(x, \hat{h}+1)) \,] + A(x)$$

$$(50) \qquad \leq \max [\, \Theta(\hat{h}-1, \hat{p}), \Theta(\hat{h}, \hat{p}-1), \Theta(\hat{h}+1, \hat{p}-W(x, \hat{h}+1)) \,] + A(x)$$

Now apply (40), (38) and (39) to (50) to reach the desired conclusion (48):

$$\Upsilon(x, \hat{h}, \hat{p}) \leq \Theta(\hat{h}, \hat{p}-1) + A(x)$$

$$= \Theta(\hat{h}, \hat{p})$$

The proof for the remaining case, viz., $\hat{h} = H(x)$, is similar, but (23) and (12) are used to show that the term $\Upsilon(x, \hat{h}+1, \hat{p}-W(x, \hat{h}+1))$ in (49) is not greater than the term $\Upsilon(x, \hat{h}, \hat{p}-1)$. The proof of this case will not be presented. This completes the proof of (44) by induction on h, thereby completing the proof of (41) by induction on p.

The desired conclusion (34) follows from definition (13), (41), and definition (37):

$$\Xi(x, p') = \Upsilon(x, H(x), p') - \Upsilon(x, 1, p') + 1$$

$$\leq \Theta(H(x), p') - \Upsilon(x, 1, p') + 1$$

$$= \left[\sum_{\underline{h}=2}^{H(x)} W(x, \underline{h})\right] \cdot A(x) + 1$$

This completes the proof of Theorem 1.

3.2 Example 1: Round Robin Scheduling

Consider a system that satisfies the assumptions of Chapter 2 and has the layout shown in Figure 1. The system includes a session x for which $H(x) \geq 2$ and $N(x) \geq 2$. Session x uses links $l_1, l_2, \ldots, l_{H(x)}$, in that order. (For each of these links, there is another link with opposite direction that is not shown in Figure 1 and is used only to return flow control permits.) Sessions $y_1, y_2, \ldots, y_{N(x)-1}$ use only link $l_{H(x)}$. Round robin link scheduling is used. The ring position for $l_{H(x)}$ at time 0 is x. The window size for buffer 1 of session x is at least two and may be either finite or infinite. The window sizes for buffers 2 through $H(x)$ of session x are at least two but finite, and these buffers are initially full. For each session $y_1, y_2, \ldots, y_{N(x)-1}$, the capacity of buffer 1 is at least two, and this buffer is initially nonempty. † Every session in the system has heavy demand; i.e.,

† In practice, this "initial" system state could arise if sessions x, y_1, $y_2, \ldots, y_{N(x)-1}$ started *before* time 0, when there were already many *other* sessions using link $l_{H(x)}$, and if these extra sessions terminated at time 0.

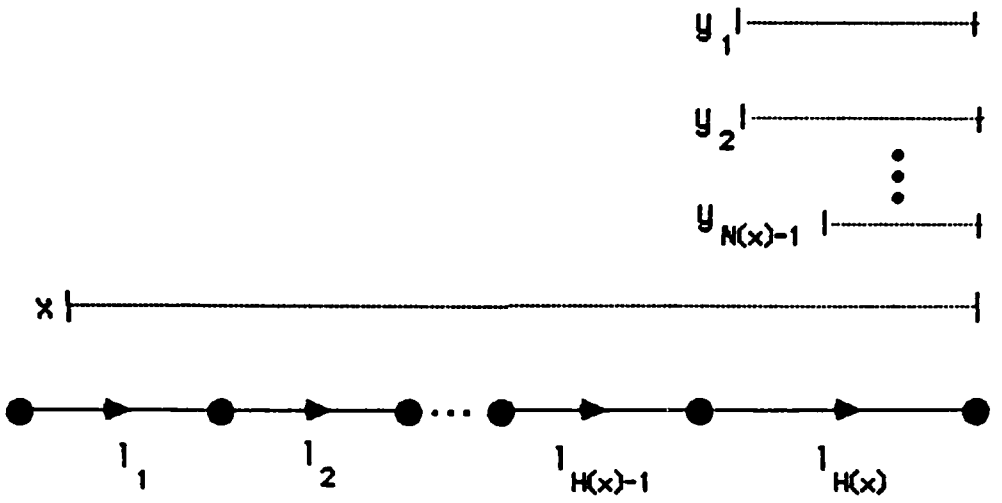


FIGURE 1

$$
\begin{aligned}
(51) \qquad C(x, 0, t-1, t) &= C(y_1, 0, t-1, t) \\
&= C(y_2, 0, t-1, t) \\
&\;\;\vdots \\
&= C(y_{N(x)-1}, 0, t-1, t) \\
&= 1
\end{aligned}
$$

for all times $t \geq 1$.

The evolution of this system is simple. Session x transmits packets over link $l_{H(x)}$ during slots $N(x)$, $2N(x)$, $3N(x)$, ... Let p' be the youngest packet of session x in buffer 2 at time 0; i.e.,

$$
(52) \qquad p' = \sum_{h=2}^{H(x)} W(x, h)
$$

Packet p' is transmitted over $l_{H(x)}$ during slot $p' \cdot N(x)$. Therefore, by definition (13),

$$
\begin{aligned}
\Xi(x, p') &= \Upsilon(x, H(x), p') - \Upsilon(x, 1, p') + 1 \\
&= p' \cdot N(x) - 0 + 1 \\
(53) \qquad &= \left[\sum_{h=2}^{H(x)} W(x, h) \right] \cdot N(x) + 1
\end{aligned}
$$

This matches the upper bound (25) of Theorem 1.

Things are slightly different for packets $p > p'$ of session x. Because of

the extreme initial conditions, the system experiences a mild transient. No permits for buffer 2 of session x are returned to hop 1 until time slot $N(x)+H(x)-2$, after which one permit is returned every $N(x)$ slots. Therefore, for each packet $p > p'$ of session x,

$$\Upsilon(x, 1, p) = [N(x) + H(x) - 2] + (p - p' - 1)\cdot N(x) + 1$$

$$= (p - p')\cdot N(x) + H(x) - 1 \tag{54}$$

Link $l_{H(x)}$, however, functions periodically even from time 0, and

$$\Upsilon(x, H(x), p) = p\cdot N(x) \tag{55}$$

By definition (13), (54), (55), and definition (52),

$$\Xi(x, p) = \Upsilon(x, H(x), p) - \Upsilon(x, 1, p) + 1$$

$$= p'\cdot N(x) - H(x) + 2$$

$$= \left(\left[\sum_{h=2}^{H(x)} W(x, h)\right]\cdot N(x) + 1\right) - [H(x) - 1] \tag{56}$$

Comparing (56) with (53) shows that the cross-network delay for each packet $p > p'$ is slightly less than the delay for packet p'.

For future reference, note that the long-term average throughput $R_A(x)$ of session x is $1/N(x)$.

3.3 Example 2: First-Come-First-Served Scheduling

Consider a system that satisfies the assumptions of Chapter 2 and has the layout shown in Figure 2. The system includes a session x for which $H(x) \geq 3$ and $N(x) \geq 2$. Session x uses links $l_1, l_2, \ldots, l_{H(x)}$, in that order. (For each of these links, there is another link with opposite direction that is not shown in Figure 2 and is used only to return flow control permits.) Sessions $y_1, y_2, \ldots, y_{N(x)-1}$ use only link $l_{H(x)-1}$. Sessions $z_1, z_2, \ldots, z_{N(x)-1}$ use only $l_{H(x)}$. The window size for buffer 1 of session x is at least two and may be either finite or infinite. The window sizes for buffers 2 through $H(x)-1$ of session x are at least two but finite, and these buffers are full at time 0. The initial level of buffer $H(x)$ of session x is one, and $W(x, H(x)) = 2$. Buffers 1 and 2 of sessions $y_1, y_2, \ldots, y_{N(x)-1}, z_1, z_2, \ldots, z_{N(x)-1}$ have capacity $W'' \geq 2$; buffer 1 for each of these sessions is initially full, and buffer 2 is initially empty. First-come-first-served link scheduling is used. Session x appears last in the tie-breaking lists of links $l_{H(x)-1}$ and $l_{H(x)}$. The transmitter queues at links $l_1, l_2, \ldots, l_{H(x)-2}$ are empty at time 0, because of a lack of permits for session x. Initially, the transmitter queue for $l_{H(x)-1}$ contains W'' reservations for each session $y_1, y_2, \ldots, y_{N(x)-1}$ (in any order) followed by one reservation for session x. (Although x has $W(x, H(x)-1)$ packets waiting to be transmitted over hop $H(x)-1$, only one of these has a permit for buffer $H(x)$. Hence x has only one reservation in the transmitter queue for $l_{H(x)-1}$.) Initially, the transmitter queue for $l_{H(x)}$ contains W''

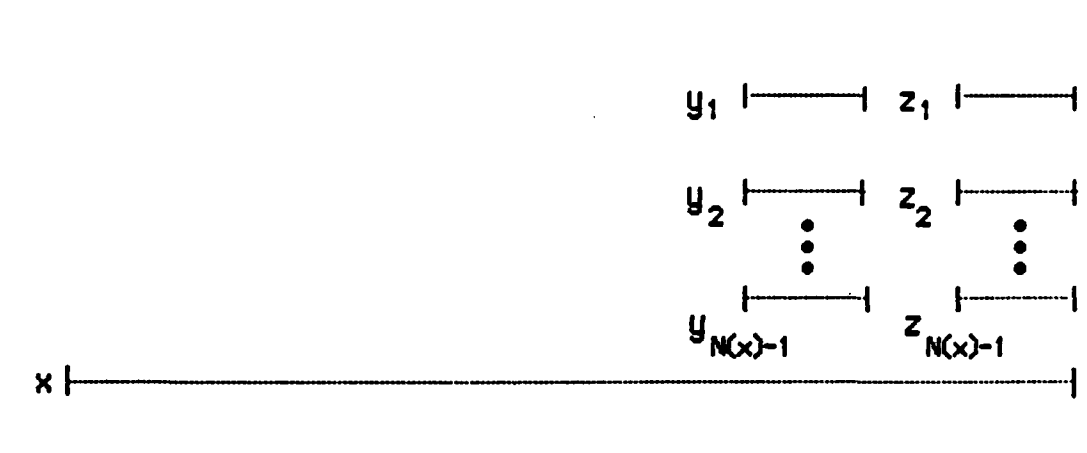


FIGURE 2

reservations for each session z_1 , z_2 , ... , $z_{N(x)-1}$ (in any order) followed by one reservation for session x. Note that the initial transmitter queue lengths at $l_{H(x)-1}$ and $l_{H(x)}$ are $N(x)\cdot W''- W''+1$. † Every session in the system has heavy demand; i.e.,

$$
\begin{aligned}
(57) \qquad C(x, 0, t-1, t) &= C(y_1 , 0, t-1, t) \\
&\ \ \vdots \\
&= C(y_{N(x)-1} , 0, t-1, t) \\
&= C(z_1 , 0, t-1, t) \\
&\ \ \vdots \\
&= C(z_{N(x)-1} , 0, t-1, t) \\
&= 1
\end{aligned}
$$

for all times $t \geq 1$.

The evolution of this system is simple. Session x transmits packets over links $l_{H(x)-1}$ and $l_{H(x)}$ during slots $[N(x)\cdot W''- W''+1]$, $2\cdot[N(x)\cdot W''- W''+1]$, $3\cdot[N(x)\cdot W''- W''+1]$, ... Let p' be the youngest

† In practice, this "initial" system state could arise if sessions x, y_1 , ... , $y_{N(x)-1}$, z_1 , ... , $z_{N(x)-1}$ started at various times *before* time 0, when there were already many *other* sessions using links $l_{H(x)-2}$, $l_{H(x)-1}$ and $l_{H(x)}$, and if these extra sessions terminated at time 0.

packet of session x in buffer 2 at time 0; i.e.,

$$p' = \sum_{h=2}^{H(x)} W(x, h) - 1 \tag{58}$$

Packet p' is transmitted over $l_{H(x)}$ during slot $p' \cdot [N(x) \cdot W'' - W'' + 1]$. Therefore, by definition (13),

$$\begin{aligned} \Xi(x, p') &= \Upsilon(x, H(x), p') - \Upsilon(x, 1, p') + 1 \\ &= p' \cdot [N(x) \cdot W'' - W'' + 1] - 0 + 1 \\ &= \left[\sum_{h=2}^{H(x)} W(x, h) - 1 \right] \cdot [N(x) \cdot W'' - W'' + 1] + 1 \end{aligned} \tag{59}$$

This is almost equal to the upper bound (25) of Theorem 1.

Things are slightly different for packets $p > p'$ of session x. Because of the extreme initial conditions, the system experiences a mild transient. No permits for buffer 2 of session x are returned to hop 1 until time slot $[N(x) \cdot W'' - W'' + 1] + H(x) - 3$, after which one permit is returned every $N(x) \cdot W'' - W'' + 1$ slots. Therefore, for each packet $p > p'$ of session x,

$$\begin{aligned} \Upsilon(x, 1, p) &= \Big[[N(x) \cdot W'' - W'' + 1] + H(x) - 3 \Big] \\ &\quad + (p - p' - 1) \cdot [N(x) \cdot W'' - W'' + 1] + 1 \\ &= (p - p') \cdot [N(x) \cdot W'' - W'' + 1] + H(x) - 2 \end{aligned} \tag{60}$$

Links $l_{H(x)-1}$ and $l_{H(x)}$, however, function periodically even from time 0.

and

$$\Upsilon(x, H(x), p) = p\cdot[N(x)\cdot W'' - W'' + 1] \tag{61}$$

By definition (13), (60), (61), and definition (58),

$$\begin{aligned}\Xi(x, p) &= \Upsilon(x, H(x), p) - \Upsilon(x, 1, p) + 1 \\ &= p'\cdot[N(x)\cdot W'' - W'' + 1] - H(x) + 3 \\ &= \left(\left[\sum_{h=2}^{H(x)} W(x, h) - 1\right]\cdot[N(x)\cdot W'' - W'' + 1] + 1\right) - [H(x) - 2] \qquad (62)\end{aligned}$$

Comparing (62) with (59) shows that the cross-network delay for each packet $p > p'$ is slightly less than the delay for packet p'.

For future reference, note that the long-term average throughput $R_A(x)$ of session x is $1/[N(x)\cdot W''- W''+1]$. †

† In this example, it is critical that $W(x, H(x)) = 2$. The long-term average throughput and the cross-network delay for session x could be significantly improved by increasing $W(x, H(x))$.

4. SESSION THROUGHPUTS IN SYSTEMS WITH LARGE WINDOWS

This chapter studies the fairness of session throughputs in systems where the window size $W(x, h)$ is finite but very large, for each buffer h of each session x in the range $1 \leq h \leq H(x)$. The exact assumptions about the window sizes vary from section to section. Round robin link scheduling is assumed throughout the chapter, except in Section 4.6, where it is shown that certain round robin results do *not* hold if first-come-first-served scheduling is used instead. This chapter assumes that each session x has a well-defined, real demand rate $\lambda(x)$ in the range $0 \leq \lambda(x) \leq 1$, † but the detailed demand assumptions vary among the sections.

The chapter is organized as follows. The max-min flow criterion, which is taken as the definition of throughput fairness throughout this chapter, is described in Section 4.1. According to this criterion, each session has a unique fair throughput rate. Section 4.2 contains some preliminary results needed in later sections. Theorem 2 of Section 4.3 analyzes a system during an interval (T_1, T_2) of smooth demand. Specifically, Theorem 2 assumes that there exists a constant Δ such that the demand of each session x over each subinterval $(s, t]$ of (T_1, T_2) is within Δ packets of the nominal amount

† An example of a session x with demand rate $\lambda(x) = 0$ is a session that has only a finite number of chances at hop 0 after time 0 and therefore injects only a finite number of packets into the network.

$\lambda(x)\cdot(t-s)$. It is also assumed that most window sizes are at least $3\cdot(H+1)^S\cdot N^{S-1}\cdot(\Delta+2)$. Theorem 2 concludes that the throughput of each session x over each subinterval of (T_1, T_2) is within $(H+1)^S\cdot N^{2S-1}\cdot(W'+3\Delta+4)$ packets of the fair amount, regardless of the length of the subinterval.

A steady-state analysis is found in Section 4.4. This section assumes that there exists a constant Δ such that the demand of each session x over each interval $(s, t]$ is within Δ packets of the nominal amount $\lambda(x)\cdot(t-s)$. Again, most window sizes are assumed to be at least $3\cdot(H+1)^S\cdot N^{S-1}\cdot(\Delta+2)$. Corollary 1 of Theorem 2 concludes that the long-term average throughput $R_A(x)$ of each session x equals its fair rate. Theorem 3, the steady-state analog of Theorem 2, shows that there exists a time $T_{SS} \geq 0$ such that the throughput of each session x over each interval later than T_{SS} is within $(H+1)^S\cdot N^{S-1}\cdot(\Delta+2)$ packets of the fair amount, regardless of the length of the interval. † Note that this bound on throughput unfairness in steady state is tighter than the transient bound of Theorem 2 and is independent of W'. Section 4.4 also contains several corollaries of Theorem 3 dealing with steady-state buffer levels.

---

† Although no upper bound is known for the length T_{SS} of the transient period, Theorem 2 shows that the throughput of each session x during the transient period is within $(H+1)^S\cdot N^{2S-1}\cdot(W'+3\Delta+4)$ packets of the fair amount.

Theorem 4 of Section 4.5 uses a burstier demand model: the sessions are assumed to have independent Bernoulli demand processes. Most window sizes are assumed to be at least $12 \cdot (H+1)^S \cdot N^{S-1}$ and at least a certain fraction α of W'. Theorem 4 concludes that (with probability one) the long-term average throughput $R_A(x)$ of each session x differs from its fair rate by no more than $\frac{74S \cdot (H+1)^{2S} \cdot N^{2S-1}}{\alpha \cdot (W')^{0.5}}$. In other words, the session throughput rates can be made arbitrarily close to the fair rates by choosing window sizes that are of the same order of magnitude and are sufficiently large.

Theorems 2, 3 and 4 show that enormous windows (of comparable size) are sufficient to guarantee fair (or nearly fair) throughput rates. One wonders whether large windows are actually necessary. Section 4.4 presents an example where the throughput rates are quite unfair unless very large windows are used. Things could be worse, however: Section 4.6 shows that if first-come-first-served link scheduling is used instead of round robin scheduling, then even large windows cannot guarantee throughput fairness. Chapter 5 determines what throughput guarantees *are* possible in systems with small windows.

4.1 Fairness Criterion

This section describes the max-min flow criterion, which will be taken as the definition of throughput fairness throughout Chapter 4. The version of the criterion presented here was proposed by Hayden [13]. A similar criterion was developed independently by Jaffe [16, 17]. Later, Gafni and Bertsekas [6] phrased the principle more economically and generalized it. The criterion is described here as it applies to the system model presented in Chapter 2. In particular, it is assumed that the sessions and routes have been specified and that all links have unit capacity. It is also assumed that each session x has a well-defined, real demand rate $\lambda(x)$ in the range $0 \leq \lambda(x) \leq 1$. †

First, let us define some terms. An *allocation* r is a function that assigns each session x a real rate $r(x)$ in the range $0 \leq r(x) \leq \lambda(x)$ without violating the link capacities. In other words, the sum of the rates for all sessions sharing a link l cannot exceed the link's capacity:

$$\sum_{x \text{ using } l} r(x) \leq 1 \tag{63}$$

The *full rate list* of an allocation r is a unique vector consisting of the rates $r(x)$ assigned to all the sessions x. If the same rate is assigned to k different

---

† It is easy to generalize the max-min flow criterion to systems where the link capacities and the session demand rates are arbitrary nonnegative real numbers.

sessions, then that rate appears k times in the full rate list. The components of the full rate list must appear in nondecreasing order. The *reduced rate list* of an allocation is formed by deleting duplicate values from the full rate list.

Now fairness can be defined. An allocation r satisfies the *max-min flow criterion* if no other allocation has a full rate list that is lexicographically greater than the full rate list of r. Roughly speaking, this means that the smallest rate assigned to any session by r is as large as possible and, subject to that constraint, the second-smallest assigned rate is as large as possible, etc. Each of these nested optimization problems can be formulated as a linear program [13], and it is not difficult to show that there exists a unique allocation that solves them all. The rates assigned in this unique max-min allocation will be called the *fair* rates. The objective of Chapter 4 is to determine conditions under which the long-term average throughput $R_A(x)$ of each session x equals its fair rate.

Let I denote the length of the reduced rate list for the max-min allocation (i.e., the number of distinct fair rates), and let $R_F(i)$ denote the i^{th} element of this list. Any session whose max-min fair rate is $R_F(i)$ is said to have *congestion index* i. For example, all sessions with the smallest fair rate have congestion index 1, and all sessions with the largest fair rate have congestion index I. Let $I(x)$ denote the congestion index of session x. In other words, the fair rate for session x is $R_F(I(x))$.

The max-min flow criterion can also be stated in terms of bottlenecks. Suppose some allocation is given (not necessarily the max-min allocation). A link l is called a *bottleneck link* for a session x using l if the rate $r(x)$ assigned to x is at least as large as the assigned rate of any other session using l, and if the entire capacity of l is assigned to the sessions using it. The following equivalence is not difficult to prove: an allocation satisfies the max-min flow criterion if and only if, for each session x, either $r(x) = \lambda(x)$ (i.e., the demand of x is a bottleneck) or x has at least one bottleneck link.

Once the max-min criterion has been stated in terms of bottlenecks, it is easy to see why round robin link scheduling might be expected to achieve the max-min fair rates [7, 9]. Consider a session whose demand exceeds its throughput. Packets for this session should accumulate at the input to its most congested link. Therefore, the session should never have to forfeit its turn in that link's round robin. This ensures that the session's average throughput will be at least as large as that of any of its competitors at that link, and it also ensures that the link will stay busy. Thus the link should be a bottleneck link for that session in the technical sense defined above. Every session that is not limited by its own demand should have such a bottleneck link; hence the resulting average throughputs should equal the max-min fair rates. Of course, this crude plausibility argument does not constitute a proof.

For the remainder of this thesis, the term "bottleneck" will be used to mean "bottleneck with respect to the max-min allocation." In other words, a link l is

a *bottleneck link* for a session x using l if

$$(64) \qquad \begin{cases} I(y) \leq I(x) & \text{for all sessions } y \text{ using } l \\ \sum\limits_{y \text{ using } l} R_F(I(y)) = 1 \end{cases}$$

If a bottleneck link for session x has hop number h, $1 \leq h \leq H(x)$, then h is a *bottleneck hop* for x. Hop 0 is said to be a bottleneck hop for x if

$$(65) \qquad R_F(I(x)) = \lambda(x)$$

i.e., if session x is bottlenecked by its demand. Every session x has at least one bottleneck hop h in the range $0 \leq h \leq H(x)$. (Hop $H(x)+1$ is never said to be a bottleneck hop.)

For future reference, let us define $R_C(x, h)$ and $R'_C(x, l)$ for a session x that uses a link l as its hop h, $1 \leq h \leq H(x)$: †

$$(66) \qquad R_C(x, h) = R'_C(x, l) = \frac{1 - \sum\limits_{y \in Y(x, l)} R_F(I(y))}{| Z(x, l) |}$$

where $Y(x, l)$ is the set of sessions y using l for which $I(y) < I(x)$, and $Z(x, l)$ is the set of sessions z using l for which $I(z) \geq I(x)$. Note that

† It will be shown in Sections 4.3.1 and 4.4.3 that if round robin scheduling is used, and if the session demands are sufficiently regular, and if the windows are large enough, then $R'_C(x, l)$ is a lower bound on the rate at which the round robin scheduler for link l offers chances to session x.

$$(67)\quad R_C(x,h) \;\geq\; \frac{\sum_{z\in Z(x,l)} R_F(I(z))}{|\,Z(x,l)\,|} \;\geq\; \frac{\sum_{z\in Z(x,l)} R_F(I(x))}{|\,Z(x,l)\,|} \;=\; R_F(I(x))$$

If h is a bottleneck hop for x, then equality holds throughout (67). If h is not a bottleneck hop for x, then one or both of the inequalities of (67) must be strict. Let us also define $R_C(x,0)$ for each session x:

$$(68)\qquad R_C(x,0) \;=\; \lambda(x)$$

Note that

$$(69)\qquad R_C(x,0) \;\geq\; R_F(I(x))$$

and that hop 0 is a bottleneck for session x if and only if equality holds in (69). In summary, for any hop h of any session x in the range $0 \leq h \leq H(x)$,

$$(70)\qquad \begin{cases} R_C(x,h) \;=\; R_F(I(x)) & \text{if } h \text{ is a bottleneck hop for } x \\ R_C(x,h) \;>\; R_F(I(x)) & \text{if } h \text{ is not a bottleneck hop for } x \end{cases}$$

The concepts of this section will now be illustrated, using the system of Figure 3 as an example. The network contains links l_1, l_2, l_3, and l_4. (For each of these links, there is another link with opposite direction that is not shown in Figure 3 and is used only to return flow control permits.) Each link has unit capacity. Session x_1 uses only link l_1. Session x_2 uses l_1 followed by l_2. Session x_3 uses all four links. Sessions x_4 and x_5 use only l_3. Sessions x_6 and x_7 use only l_4. Every session has a demand rate of 1, except session x_4, whose demand rate is $1/6$, and session x_6, whose demand rate is $1/3$. The max-min fair rate for each session is $1/3$, except session x_4, whose fair rate is

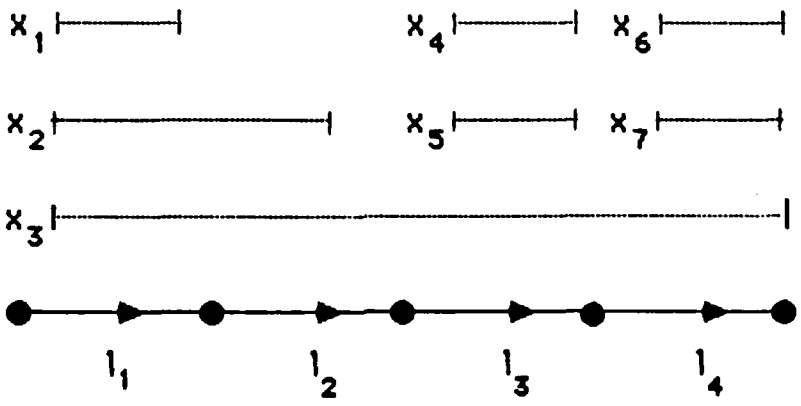


FIGURE 3

$1/6$, and session x_5 , whose fair rate is $1/2$. The full rate list for the max-min allocation is ($1/6$, $1/3$, $1/3$, $1/3$, $1/3$, $1/3$, $1/2$) . The reduced rate list for the max-min allocation is ($1/6$, $1/3$, $1/2$) . Session x_4 has congestion index 1. Session x_5 has congestion index 3. The other sessions have congestion index 2. Sessions x_1 and x_2 have link l_1 as a bottleneck. Session x_3 has two bottleneck links -- l_1 and l_4 . The bottleneck hop for x_4 is hop 0 -- its demand. Link l_3 is the bottleneck for x_5 . Session x_6 has its demand and l_4 as bottlenecks. Link l_4 is also the bottleneck for x_7 . Link l_2 is not a bottleneck for any session, since it has unused capacity.

Note that, in this example, the max-min allocation does not maximize the sum of the session rates or minimize idle link capacity. The max-min flow criterion is certainly not the only reasonable capacity allocation strategy one could propose. For some applications, one might be willing to tolerate some unfairness in order to achieve more efficiency.

4.2 Preliminary Results

This section contains various lemmas needed for Sections 4.3 and 4.4. Lemma 2 (Section 4.2.2) assumes an upper bound on the throughput of sessions whose congestion index is less than a particular value i; this bound is used to derive a lower bound on the number of chances offered by a link's round robin scheduler to a session with congestion index i. Lemma 6 assumes lower bounds on the demand of a particular session and on the number of chances the session receives at each link in its path; these bounds and the assumption of large windows are used to derive a lower bound on the throughput of the session. The proof of Lemma 6 requires Lemmas 3, 4 and 5; all four of these are found in Section 4.2.3. Lemma 7 (Section 4.2.4) assumes an upper bound on the demand of a particular session x and a lower bound on the throughput of sessions whose congestion indices are $I(x)$ or less; these bounds and the properties of max-min fairness (viz., the existence of bottleneck hops) are used to derive an upper bound on the throughput of session x. Lemma 1 (Section 4.2.1) notes various inequalities relating the functions $E_{CL}(\Delta, i)$, $E_{PL}(\Delta, i)$, $E_{PU}(\Delta, i)$, $F_{CL}(\Delta, i)$, $F_{PL}(\Delta, i)$, $F_{PU}(\Delta, i)$, $F''_{PU}(\Delta, i)$, $D_{CL}(\Delta, i, K)$, $D_{PL}(\Delta, i, K)$, and $D_{PU}(\Delta, i, K)$ defined below. The argument Δ is a real number, and i and K are integers. (The subscripts C, P, L, and U stand for "chances," "packets," "lower bound," and "upper bound," respectively. The subscripts indicate how the functions will be used.)

$$(71)\quad E_{CL}(\Delta, i) = \begin{cases} (H+1)^{i-1}\cdot(N-1)\cdot N^{2i-3}\cdot(W'+2\Delta+2) & \text{for } i \geq 2 \\ 0 & \text{for } i \leq 1 \end{cases}$$

$$(72)\quad E_{PL}(\Delta, i) = \begin{cases} (H+1)^{i}\cdot(N-1)\cdot N^{2i-3}\cdot(W'+2\Delta+2) & \text{for } i \geq 2 \\ 0 & \text{for } i \leq 1 \end{cases}$$

$$(73)\quad E_{PU}(\Delta, i) = \begin{cases} (H+1)^{i}\cdot N^{2i-1}\cdot(W'+2\Delta+2) & \text{for } i \geq 1 \\ 0 & \text{for } i \leq 0 \end{cases}$$

$$(74)\quad F_{CL}(\Delta, i) = \begin{cases} (H+1)^{i-1}\cdot N^{i-1}\cdot(\Delta+2) - 1 & \text{for } i \geq 1 \\ 0 & \text{for } i \leq 0 \end{cases}$$

$$(75)\quad F_{PL}(\Delta, i) = \begin{cases} (H+1)^{i}\cdot N^{i-1}\cdot(\Delta+2) - 1 & \text{for } i \geq 1 \\ 0 & \text{for } i \leq 0 \end{cases}$$

$$(76)\quad F_{PU}(\Delta, i) = \begin{cases} (H+1)^{i}\cdot N^{i-1}\cdot(\Delta+2) & \text{for } i \geq 1 \\ 0 & \text{for } i \leq 0 \end{cases}$$

$$(77)\quad F''_{PU}(\Delta, i) = \begin{cases} (H+1)^{i}\cdot N^{i}\cdot(W'+2\Delta+2) & \text{for } i \geq 1 \\ 0 & \text{for } i \leq 0 \end{cases}$$

$$(78)\quad D_{CL}(\Delta, i, K) = E_{CL}(\Delta, i) + K\cdot F_{CL}(\Delta, i)$$

$$(79)\quad D_{PL}(\Delta, i, K) = E_{PL}(\Delta, i) + K\cdot F_{PL}(\Delta, i)$$

$$(80)\quad D_{PU}(\Delta, i, K) = E_{PU}(\Delta, i) + K\cdot F_{PU}(\Delta, i)$$

4.2.1 Lemma 1: Miscellaneous Inequalities

The following inequalities hold for all *positive* integers i and K and all *nonnegative* real numbers Δ .

(81) $F_{PL}(\Delta, i) \geq 0$

(82) $D_{PL}(\Delta, i, K-1) \geq 0$

(83) $F_{PU}(\Delta, i-1) \geq 0$

(84) $D_{PU}(\Delta, i-1, K) \geq 0$

(85) $F_{PL}(\Delta, i+1) \geq F_{PL}(\Delta, i)$

(86) $D_{PL}(\Delta, i+1, K) \geq D_{PL}(\Delta, i, K)$

(87) $F_{PU}(\Delta, i+1) \geq F_{PU}(\Delta, i)$

(88) $D_{PU}(\Delta, i+1, K) \geq D_{PU}(\Delta, i, K)$

(89) $F_{CL}(\Delta, i) \geq \Delta$

(90) $D_{CL}(\Delta, i, K) \geq K \cdot \Delta$

(91) $F_{CL}(\Delta, i) \geq (N-1) \cdot F_{PU}(\Delta, i-1) + 1$

(92) $D_{CL}(\Delta, i, K) \geq (N-1) \cdot D_{PU}(\Delta, i-1, K) + K$

(93) $F_{PL}(\Delta, i) \geq (H+1) \cdot F_{CL}(\Delta, i) + H$

(94) $D_{PL}(\Delta, i, K) \geq (H+1) \cdot D_{CL}(\Delta, i, K) + K \cdot H$

(95) $F_{PU}(\Delta, i) \geq F_{PL}(\Delta, i) + 1$

(96) $F''_{PU}(\Delta, i) \geq (N-1) \cdot F_{PL}(\Delta, i) + W' \cdot H + \Delta$

(97) $D_{PU}(\Delta, i, K) \geq (N-1) \cdot D_{PL}(\Delta, i, 1) + D_{PL}(\Delta, i, K-1) + W' \cdot H + \Delta$

These inequalities follow directly from definitions (71) - (80); the proofs will not be presented.

4.2.2 Lemma 2: Lower Bound on Chances, given Upper Bound on Throughput

Suppose a system has been specified that satisfies the assumptions of Chapter 2 and uses round robin scheduling. Suppose each session has a well-defined demand rate. Let x be some session, and let l be some link used by x. Let K be a positive integer, and let s_1 , t_1 , s_2 , t_2 , ... , s_K , t_K be times satisfying $0 \leq s_1 \leq t_1 \leq s_2 \leq t_2 \leq \cdots \leq s_K \leq t_K$. Let

$$G \geq 0 \tag{98}$$

be a real number such that, for each session y using l with $I(y) < I(x)$,

$$\sum_{k=1}^{K} P'(y, l, s_k , t_k) \leq R_F(I(y)) \cdot \sum_{k=1}^{K} (t_k - s_k) + G \tag{99}$$

It follows that

$$\sum_{k=1}^{K} C'(x, l, s_k , t_k) \geq R'_C(x, l) \cdot \sum_{k=1}^{K} (t_k - s_k) - (N - 1) \cdot G - K \tag{100}$$

Proof of Lemma 2

Let Y denote the (possibly empty) set of sessions y on l for which $I(y) < I(x)$. Let Z denote the set of sessions z on l for which $I(z) \geq I(x)$. Note that Z includes x. For $k = 1, 2, \ldots , K$, let q_k be the number of slots in $(s_k , t_k]$ that are not used by sessions in Y:

$$q_k = (t_k - s_k) - \sum_{y \in Y} P'(y, l, s_k, t_k) \tag{101}$$

Since these q_k slots are not used by Y, the round robin scheduler at l will offer each of these slots to at least one session in Z; hence

$$q_k \leq \sum_{z \in Z} C'(z, l, s_k, t_k)$$

$$\leq |Z| \cdot \max_{z \in Z} C'(z, l, s_k, t_k) \tag{102}$$

By the operating rules of the round robin scheduler, session x must receive almost as many chances as any other session on l during $(s_k, t_k]$; in particular,

$$C'(x, l, s_k, t_k) \geq \max_{z \in Z} C'(z, l, s_k, t_k) - 1 \tag{103}$$

Combining (103), (102) and (101) yields:

$$C'(x, l, s_k, t_k) \geq \frac{1}{|Z|} \cdot \left[(t_k - s_k) - \sum_{y \in Y} P'(y, l, s_k, t_k) \right] - 1 \tag{104}$$

Summing over k yields:

$$\sum_{k=1}^{K} C'(x, l, s_k, t_k) \geq \frac{1}{|Z|} \cdot \left[\sum_{k=1}^{K} (t_k - s_k) - \sum_{y \in Y} \sum_{k=1}^{K} P'(y, l, s_k, t_k) \right] - K \tag{105}$$

Since $I(y) < I(x)$ for all y in Y, assumption (99) can be substituted above to yield:

$$\sum_{k=1}^{K} C'(x, l, s_k, t_k)$$

$$\geq \frac{1}{|Z|} \cdot \left[\sum_{k=1}^{K} (t_k - s_k) - \sum_{y \in Y} \left(R_F(I(y)) \cdot \sum_{k=1}^{K} (t_k - s_k) + G \right) \right] - K$$

$$= \frac{1}{|Z|} \cdot \left[1 - \sum_{y \in Y} R_F(I(y)) \right] \cdot \sum_{k=1}^{K} (t_k - s_k) - \frac{|Y|}{|Z|} \cdot G - K \tag{106}$$

Applying definition (66) and assumption (98) gives the desired result (100):

$$\sum_{k=1}^{K} C'(x, l, s_k, t_k) \geq R'_C(x, l) \cdot \sum_{k=1}^{K} (t_k - s_k) - \frac{|Y|}{|Z|} \cdot G - K$$

$$\geq R'_C(x, l) \cdot \sum_{k=1}^{K} (t_k - s_k) - |Y| \cdot G - K$$

$$\geq R'_C(x, l) \cdot \sum_{k=1}^{K} (t_k - s_k) - (N - 1) \cdot G - K$$

This completes the proof of Lemma 2.

4.2.3 Tandem Queues with Finite Buffers

This section derives a lower bound on the throughput of a session from given lower bounds on the session's demand and on the number of chances the session receives at each link in its path. The problem is difficult because the session's buffers, while large, are finite. The problem is solvable, however, because the session's demand and chance processes are fairly smooth.

Let us begin with slightly oversimplified sketches of the results of this section and their proofs. The key result is Lemma 3. This lemma focuses on a particular buffer h of the given session x in the range $1 \leq h \leq H(x)$. A lower bound is assumed for the throughput over hop $h-1$ (i.e., *into* buffer h) during any interval when buffer h is not full. This bound has a special form. It is the product of a nominal rate r and the length of the interval, minus a constant error. A similar lower bound, with the same nominal rate r, is assumed for the throughput over hop h (i.e., *out* of buffer h) during any interval when buffer h is not empty. In other words, it is known how the subpaths upstream and downstream of buffer h behave when isolated from each other. The capacity $W(x, h)$ of buffer h is assumed to be at least slightly larger than the sum of the error constants in the throughput bounds for the two subpaths. Lemma 3 uses the given throughput bounds for the subpaths in isolation to derive lower bounds of the same form that apply during any interval, regardless of the level of buffer h. The error constants in these bounds for the integrated system are only slightly larger than the sum of

the error constants in the bounds for the isolated subpaths.

The proof of Lemma 3 is structured as follows. To bound the throughput over hop $h-1$ during an interval $(s, t]$, this interval is divided into various subintervals. There is an initial subinterval in which buffer h moves from its initial value to nearly empty. There is a final subinterval in which buffer h moves from nearly empty to its final value. Between the initial and final subintervals are subintervals that alternate between two types. During a type 1 subinterval, buffer h is not empty, and its level moves from nearly empty to nearly full. During a type 2 subinterval, buffer h is not full, and its level moves from nearly full to nearly empty. During a type 2 subinterval, the given lower bound applies to the throughput of the isolated upstream subpath. During a type 1 subinterval, the given lower bound applies to the throughput of the isolated downstream subpath. Moreover, the throughput of the upstream subpath during a type 1 subinterval must exceed the throughput of the downstream subpath by approximately $W(x, h)$ packets, since buffer h fills during a type 1 subinterval. The window size $W(x, h)$ is large enough that the throughput excess (over the nominal amount) for the upstream subpath during a type 1 subinterval balances the possible throughput deficit (from the nominal amount) for the upstream subpath during a type 2 subinterval. Thus the throughput over hop $h-1$ during a type 1/type 2 cycle is at least r times the combined length of the two subintervals. Net deficits from the nominal throughput can only accrue during the initial and final

subintervals of the interval $(s, t]$. Hence the net throughput deficit at hop $h-1$ over the entire interval $(s, t]$ cannot be too great. Lemma 3 analyzes the throughput over hop h in a similar manner.

Lemma 4 is a simple corollary of Lemma 3. Lemma 4 assumes a lower bound on the number of chances received by a particular session x at any hop of its path -- including hop 0, the demand hop. This bound is the product of a nominal rate r and the length of time involved, minus a constant error. Sufficiently large windows are also assumed. For each buffer h of x in the range $1 \leq h \leq H(x)+1$, Lemma 4 proves the following property: during an interval when buffer h is not full, the throughput into buffer h is at least r times the length of the interval, minus a constant error. This error constant is only slightly larger than the sum of the error constants in the given chance bounds for hops 0 through $h-1$. The proof of Lemma 4 is by forward induction on h, using Lemma 3 to add successive hops to a growing upstream subpath.

Under the same assumptions as Lemma 4, Lemma 5 derives a lower bound on the throughput *out* of each buffer h of x during intervals when the buffer is not empty. The proof of Lemma 5 is by backward induction on h, using Lemma 3 to add successive hops to a growing downstream subpath.

Lemma 6 makes the same assumptions as Lemmas 4 and 5. Since Lemma 4 analyzes the subpath upstream of any buffer h when that buffer is not full, and Lemma 5 analyzes the downstream subpath when the buffer is not empty,

Lemma 6 can invoke Lemmas 4, 5, *and* 3 to derive a lower bound on the throughput of session x at any hop during any interval, regardless of the buffer levels. As usual, the bound is the product of r and the length of the interval, minus a constant error. This error constant is only slightly larger than the sum of the error constants in the given chance bounds for hops 0 through $H(x)$. Of Lemmas 3, 4, 5, and 6, only Lemma 6 is used in later sections.

4.2.3.1 Lemma 3: Lower Bound on Throughput of Concatenated Subpaths

Suppose a system has been specified that satisfies the assumptions of Chapter 2 and uses any scheduling discipline. Let x be some session. Let h be some hop of x in the range $1 \leq h \leq H(x)$. Let K be a positive integer, and let $s_1, t_1, s_2, t_2, \ldots, s_K, t_K$ be times satisfying $0 \leq s_1 \leq t_1 \leq s_2 \leq t_2 \leq \cdots \leq s_K \leq t_K$. Assume that the following two properties hold for some real numbers r, e', e'', f', and f'':

(107) If, for $k = 1, 2, \ldots, K$, $J(k)$ is any positive integer, and $v_k^{J(k)}, u_k^{J(k)-1}, v_k^{J(k)-1}, u_k^{J(k)-2}, \ldots, v_k^1, u_k^0$ are any times such that $s_k \leq v_k^{J(k)} \leq u_k^{J(k)-1} \leq v_k^{J(k)-1} \leq u_k^{J(k)-2} \leq \cdots \leq v_k^1 \leq u_k^0 \leq t_k$ and such that $B(x, h, \tau) < W(x, h)$ for all τ in $\bigcup_{j=1}^{J(k)} [v_k^j, u_k^{j-1})$, then

$$\sum_{k=1}^{K} \sum_{j=1}^{J(k)} P(x, h-1, v_k^j, u_k^{j-1}) \geq r \cdot \sum_{k=1}^{K} \sum_{j=1}^{J(k)} (u_k^{j-1} - v_k^j) - e' - \left[\sum_{k=1}^{K} J(k) \right] \cdot f'$$

(108) If, for $k = 1, 2, \ldots, K$, $J(k)$ is any positive integer, and $u_k^{J(k)}, v_k^{J(k)}, u_k^{J(k)-1}, v_k^{J(k)-1}, \ldots, u_k^1, v_k^1$ are any times such that $s_k \leq u_k^{J(k)} \leq v_k^{J(k)} \leq u_k^{J(k)-1} \leq v_k^{J(k)-1} \leq \cdots \leq u_k^1 \leq v_k^1 \leq t_k$ and such that $B(x, h, \tau) > 0$ for all τ in $\bigcup_{j=1}^{J(k)} [u_k^j, v_k^j)$, then

$$\sum_{k=1}^{K} \sum_{j=1}^{J(k)} P(x, h, u_k^j, v_k^j) \geq r \cdot \sum_{k=1}^{K} \sum_{j=1}^{J(k)} (v_k^j - u_k^j) - e'' - \left[\sum_{k=1}^{K} J(k) \right] \cdot f''$$

Also assume that

$$f' + f'' + 2 \leq W(x, h) < \infty \tag{109}$$

It follows that

$$\sum_{k=1}^{K} P(x, h-1, s_k , t_k) \geq r \cdot \sum_{k=1}^{K} (t_k - s_k) - e - K \cdot f \tag{110}$$

and

$$\sum_{k=1}^{K} P(x, h, s_k , t_k) \geq r \cdot \sum_{k=1}^{K} (t_k - s_k) - e - K \cdot f \tag{111}$$

where

$$e = e' + e'' \tag{112}$$

$$f = f' + f'' + 1 \tag{113}$$

Proof of Lemma 3

Inequality (110) will be proved first. For each k, $1 \leq k \leq K$, let us analyze the time interval $[s_k , t_k]$ separately. The first step is to break $[s_k , t_k]$ into various subintervals. Determine a positive integer $J(k)$ and define times $u_k^0 , v_k^1 , u_k^1 , v_k^2 , u_k^2 , \ldots , v_k^{J(k)} , u_k^{J(k)}$ by the procedure specified below. Examples are shown in Figures 4 and 5.

$j \leftarrow 0$

$u_k^0 \leftarrow t_k$

E: $j \leftarrow j + 1$

$v_k^j \leftarrow$ earliest time v in $[s_k\ ,\ u_k^{j-1}]$ that satisfies $B(x, h, \tau) < W(x, h)$ for all τ in $[v,\ u_k^{j-1})$

$u_k^j \leftarrow$ earliest time u in $[s_k\ ,\ v_k^j]$ that satisfies $B(x, h, \tau) > 0$ for all τ in $[u,\ v_k^j)$

if $u_k^j > s_k$ then go to **E**

$J(k) \leftarrow j$

It is not difficult to verify that u_k^j and v_k^j are well-defined and that this procedure terminates. Let us make some remarks about u_k^j and v_k^j :

(114) $s_k = u_k^{J(k)} \leq v_k^{J(k)} \leq u_k^{J(k)-1} \leq v_k^{J(k)-1} \leq \cdots \leq u_k^1 \leq v_k^1 \leq u_k^0 = t_k$

(115) $B(x, h, \tau) < W(x, h)$ for all τ in $[v_k^j\ ,\ u_k^{j-1})\ ,\quad 1 \leq j \leq J(k)$

(116) $B(x, h, v_k^j) \geq W(x, h) - 1$ for $1 \leq j \leq J(k)-1$

(Note: Strict inequality in (116) occurs only for $j = 1$ and only if $B(x, h, t_k - 1) = B(x, h, t_k) = W(x, h)$.)

(117) If $u_k^{J(k)} < v_k^{J(k)}$, then $B(x, h, v_k^{J(k)}) \geq W(x, h) - 1$

(118) $B(x, h, \tau) > 0$ for all τ in $[u_k^j\ ,\ v_k^j)\ .\quad 1 \leq j \leq J(k)$

(119) $B(x, h, u_k^j) = 1$ for $1 \leq j \leq J(k)-1$

Now the facts above will be used to analyze the throughput over the

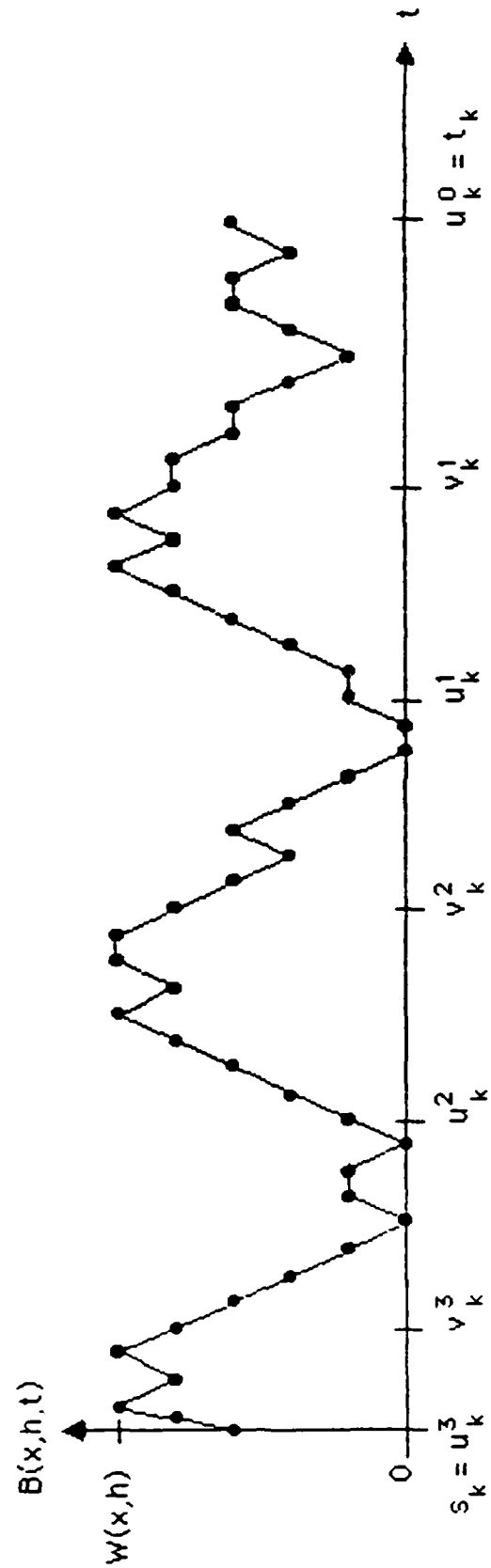


FIGURE 4

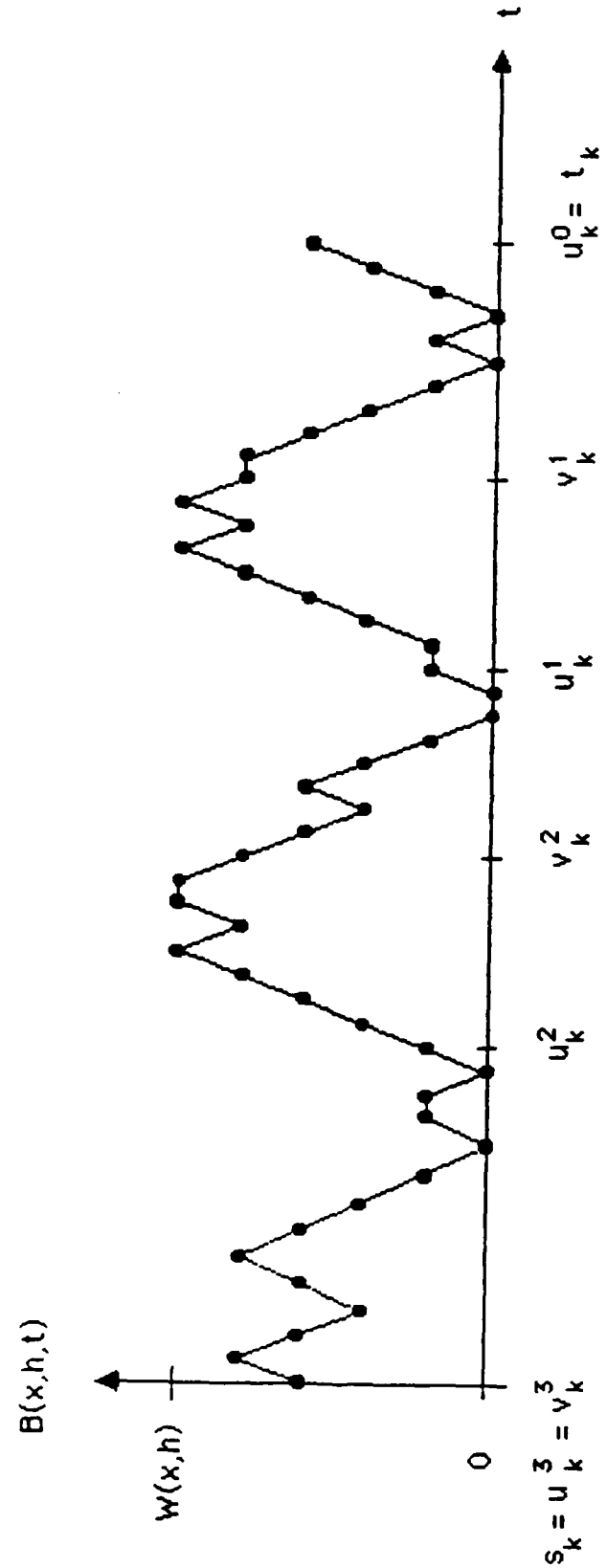


FIGURE 5

subintervals $(u_k^j , v_k^j]$. From (116), (119) and (10), it follows that

(120)
$$P(x, h-1, u_k^j , v_k^j) \geq P(x, h, u_k^j , v_k^j) + W(x, h) - 2 \quad \text{for } 1 \leq j \leq J(k)-1$$

To develop a similar inequality for $j = J(k)$, first let us justify the following claim:

(121)
$$B(x, h, u_k^{J(k)}) \leq B(x, h, v_k^{J(k)}) + 1$$

If $u_k^{J(k)} = v_k^{J(k)}$, then (121) is obviously true. If $u_k^{J(k)} < v_k^{J(k)}$, then (121) follows from (117). From (121) and (10), it follows that

(122)
$$P(x, h-1, u_k^{J(k)} , v_k^{J(k)}) \geq P(x, h, u_k^{J(k)} , v_k^{J(k)}) - 1$$

Next, the throughput over the entire interval $(s_k , t_k]$ can be studied. By (114),

$$P(x, h-1, s_k , t_k) = P(x, h-1, u_k^{J(k)} , u_k^0)$$

(123)
$$= \left[\sum_{j=1}^{J(k)} P(x, h-1, u_k^j , v_k^j)\right] + \left[\sum_{j=1}^{J(k)} P(x, h-1, v_k^j , u_k^{j-1})\right]$$

Applying (120) and (122) to (123) yields:

(124)
$$P(x, h-1, s_k , t_k) \geq \left[\sum_{j=1}^{J(k)} P(x, h, u_k^j , v_k^j)\right] + \left[\sum_{j=1}^{J(k)} P(x, h-1, v_k^j , u_k^{j-1})\right]$$
$$+ [J(k) - 1]\cdot[W(x, h) - 2] - 1$$

Finally, the throughput over the collection of intervals $(s_1 , t_1] , \ldots , (s_K , t_K]$

can be examined. Summing (124) over k yields:

$$(125)\quad \sum_{k=1}^{K} P(x,\, h-1,\, s_k\,,\, t_k) \;\geq\; \left[\sum_{k=1}^{K}\sum_{j=1}^{J(k)} P(x,\, h,\, u_k^j\,,\, v_k^j)\right] + \left[\sum_{k=1}^{K}\sum_{j=1}^{J(k)} P(x,\, h-1,\, v_k^j\,,\, u_k^{j-1})\right] + \left[\sum_{k=1}^{K}[J(k)-1]\right]\cdot[W(x,\, h)-2] \;-\; K$$

The hypotheses of the lemma can now be used to bound the right-hand side of (125). It follows from (114), (118), and assumption (108) that

$$(126)\quad \sum_{k=1}^{K}\sum_{j=1}^{J(k)} P(x,\, h,\, u_k^j\,,\, v_k^j) \;\geq\; r\cdot\left[\sum_{k=1}^{K}\sum_{j=1}^{J(k)}(v_k^j - u_k^j)\right] - e'' - \left[\sum_{k=1}^{K} J(k)\right]\cdot f''$$

Similarly, it follows from (114), (115), and assumption (107) that

$$(127)\quad \sum_{k=1}^{K}\sum_{j=1}^{J(k)} P(x,\, h-1,\, v_k^j\,,\, u_k^{j-1}) \;\geq\; r\cdot\left[\sum_{k=1}^{K}\sum_{j=1}^{J(k)}(u_k^{j-1} - v_k^j)\right] - e' - \left[\sum_{k=1}^{K} J(k)\right]\cdot f'$$

Substituting (126) and (127) into (125) yields:

$$\sum_{k=1}^{K} P(x, h-1, s_k, t_k) \geq r \cdot \left[\sum_{k=1}^{K} \sum_{j=1}^{J(k)} (u_k^{j-1} - u_k^j) \right] - (e' + e'') - K$$

$$- \left[\sum_{k=1}^{K} J(k) \right] \cdot (f' + f'')$$

$$+ \left[\sum_{k=1}^{K} [J(k) - 1] \right] \cdot [W(x, h) - 2]$$

$$= r \cdot \sum_{k=1}^{K} (u_k^0 - u_k^{J(k)}) - (e' + e'') - K \cdot (f' + f'' + 1) \tag{128}$$

$$+ \left[\sum_{k=1}^{K} [J(k) - 1] \right] \cdot [W(x, h) - f' - f'' - 2]$$

Applying definitions (112) and (113), assumption (109), and (114) yields the desired result:

$$\sum_{k=1}^{K} P(x, h-1, s_k, t_k) \geq r \cdot \sum_{k=1}^{K} (u_k^0 - u_k^{J(k)}) - e - K \cdot f$$

$$= r \cdot \sum_{k=1}^{K} (t_k - s_k) - e - K \cdot f$$

This completes the proof of inequality (110).

Now (111) will be proved. For convenience, let us rephrase assumptions (107) and (108) by juggling the superscripts of v_k^j and u_k^j :

(129) If, for $k = 1, 2, \ldots, K$, $J(k)$ is any positive integer, and $v_k^{J(k)}, u_k^{J(k)}, v_k^{J(k)-1}, u_k^{J(k)-1}, \ldots, v_k^1, u_k^1$ are any times such that $s_k \le v_k^{J(k)} \le u_k^{J(k)} \le v_k^{J(k)-1} \le u_k^{J(k)-1} \le \cdots \le v_k^1 \le u_k^1 \le t_k$ and such that $B(x, h, \tau) < W(x, h)$ for all τ in $\bigcup_{j=1}^{J(k)} [v_k^j, u_k^j)$, then

$$\sum_{k=1}^{K} \sum_{j=1}^{J(k)} P(x, h-1, v_k^j, u_k^j) \ge r \cdot \sum_{k=1}^{K} \sum_{j=1}^{J(k)} (u_k^j - v_k^j) - e' - \left[\sum_{k=1}^{K} J(k) \right] \cdot f'$$

(130) If, for $k = 1, 2, \ldots, K$, $J(k)$ is any positive integer, and $u_k^{J(k)}, v_k^{J(k)-1}, u_k^{J(k)-1}, v_k^{J(k)-2}, \ldots, u_k^1, v_k^0$ are any times such that $s_k \le u_k^{J(k)} \le v_k^{J(k)-1} \le u_k^{J(k)-1} \le v_k^{J(k)-2} \le \cdots \le u_k^1 \le v_k^0 \le t_k$ and such that $B(x, h, \tau) > 0$ for all τ in $\bigcup_{j=1}^{J(k)} [u_k^j, v_k^{j-1})$, then

$$\sum_{k=1}^{K} \sum_{j=1}^{J(k)} P(x, h, u_k^j, v_k^{j-1}) \ge r \cdot \sum_{k=1}^{K} \sum_{j=1}^{J(k)} (v_k^{j-1} - u_k^j) - e'' - \left[\sum_{k=1}^{K} J(k) \right] \cdot f''$$

The proof of (111) is similar to that of (110). As before, for $k = 1, 2, \ldots, K$, the interval $[s_k, t_k]$ must be broken into subintervals. Determine a positive integer $J(k)$ and define times $v_k^0, u_k^1, v_k^1, u_k^2, v_k^2, \ldots, u_k^{J(k)}, v_k^{J(k)}$ by the procedure specified below. Examples are shown in Figures 6 and 7.

$j \leftarrow 0$

$v_k^0 \leftarrow t_k$

E: $j \leftarrow j + 1$

$u_k^j \leftarrow$ earliest time u in $[s_k , v_k^{j-1}]$ that satisfies $B(x, h, \tau) > 0$ for all τ in $[u, v_k^{j-1})$

$v_k^j \leftarrow$ earliest time v in $[s_k , u_k^j]$ that satisfies $B(x, h, \tau) < W(x, h)$ for all τ in $[v, u_k^j)$

if $v_k^j > s_k$ then go to **E**

$J(k) \leftarrow j$

The following properties are analogous to (114) - (119):

(131) $s_k = v_k^{J(k)} \leq u_k^{J(k)} \leq v_k^{J(k)-1} \leq u_k^{J(k)-1} \leq \cdots \leq v_k^1 \leq u_k^1 \leq v_k^0 = t_k$

(132) $B(x, h, \tau) > 0$ for all τ in $[u_k^j , v_k^{j-1})$, $1 \leq j \leq J(k)$

(133) $B(x, h, u_k^j) \leq 1$ for $1 \leq j \leq J(k)-1$

(Note: Strict inequality in (133) occurs only for $j = 1$ and only if $B(x, h, t_k - 1) = B(x, h, t_k) = 0$.)

(134) If $v_k^{J(k)} < u_k^{J(k)}$, then $B(x, h, u_k^{J(k)}) \leq 1$

(135) $B(x, h, \tau) < W(x, h)$ for all τ in $[v_k^j , u_k^j)$, $1 \leq j \leq J(k)$

(136) $B(x, h, v_k^j) = W(x, h) - 1$ for $1 \leq j \leq J(k)-1$

The following inequalities can be proved by reasoning similar to that behind (120) and (122):

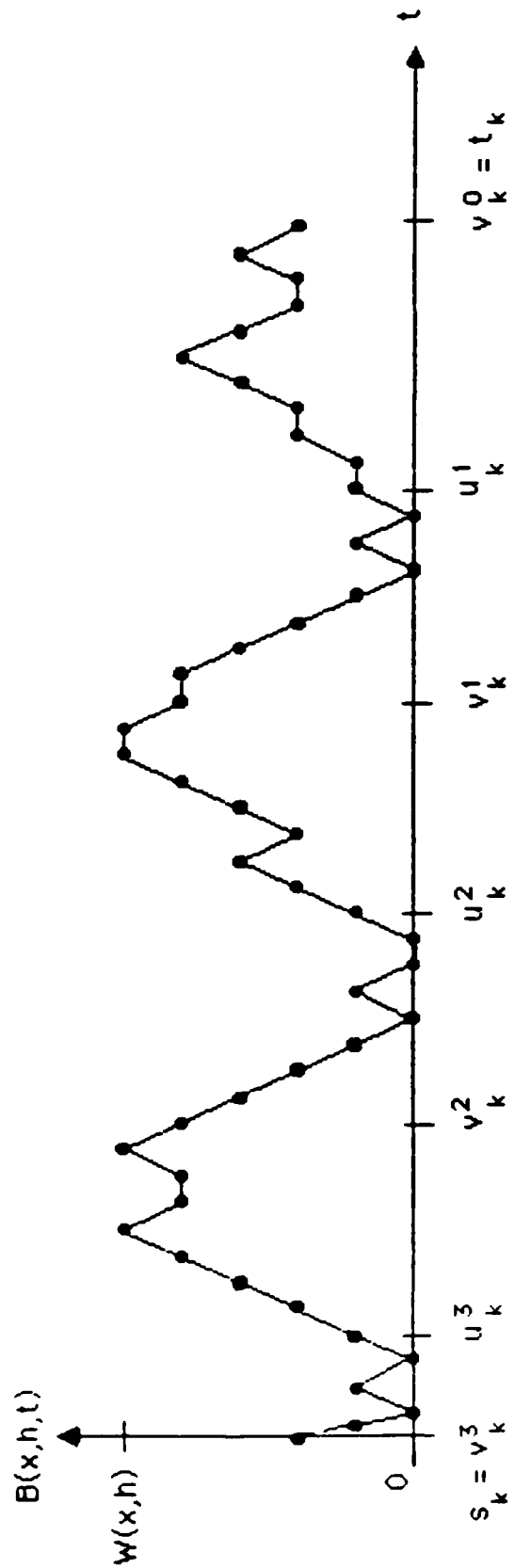


FIGURE 6

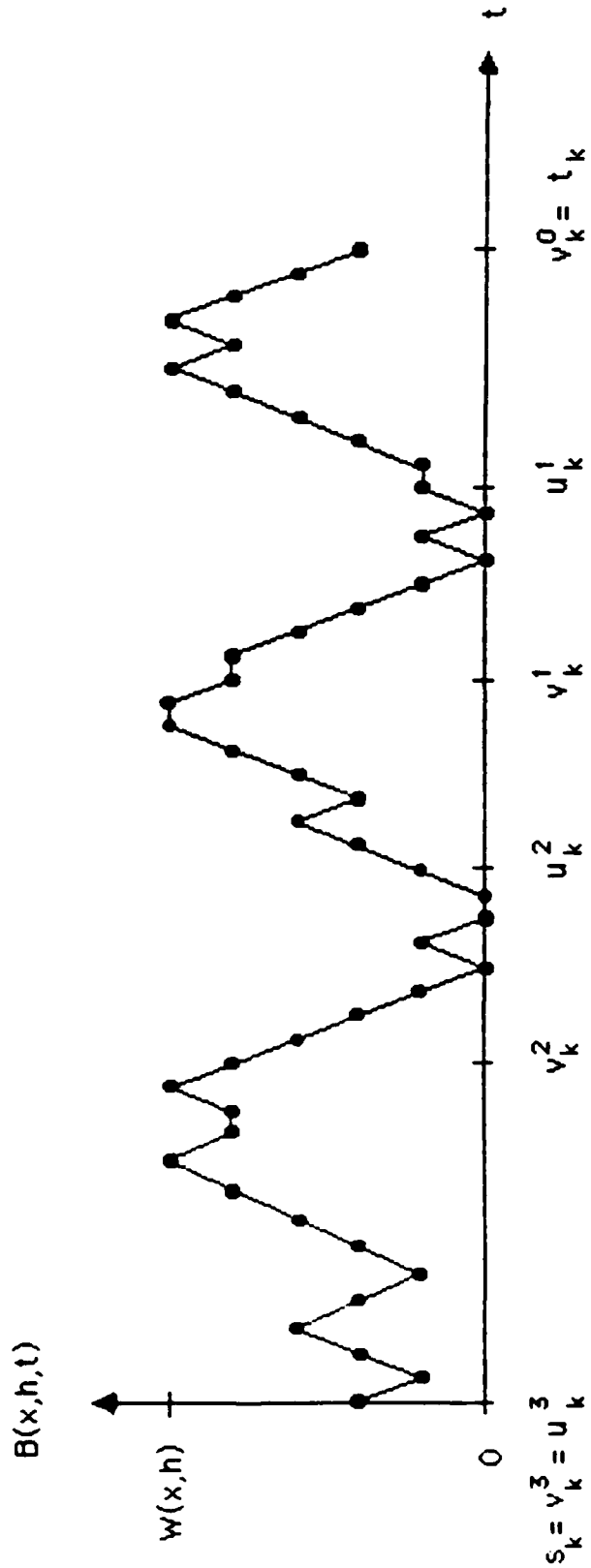


FIGURE 7

(137)
$$P(x, h, v_k^j, u_k^j) \geq P(x, h-1, v_k^j, u_k^j) + W(x, h) - 2 \quad \text{for } 1 \leq j \leq J(k)-1$$

(138)
$$P(x, h, v_k^{J(k)}, u_k^{J(k)}) \geq P(x, h-1, v_k^{J(k)}, u_k^{J(k)}) - 1$$

It follows from (131), (137), and (138) that

$$\sum_{k=1}^{K} P(x, h, s_k, t_k) = \left[\sum_{k=1}^{K}\sum_{j=1}^{J(k)} P(x, h, v_k^j, u_k^j)\right] + \left[\sum_{k=1}^{K}\sum_{j=1}^{J(k)} P(x, h, u_k^j, v_k^{j-1})\right]$$

(139)
$$\geq \left[\sum_{k=1}^{K}\sum_{j=1}^{J(k)} P(x, h-1, v_k^j, u_k^j)\right] + \left[\sum_{k=1}^{K}\sum_{j=1}^{J(k)} P(x, h, u_k^j, v_k^{j-1})\right] + \left[\sum_{k=1}^{K} [J(k) - 1]\right] \cdot [W(x, h) - 2] - K$$

Because of (131), (135), and (132), assumptions (129) and (130) can be applied to (139) to yield:

$$\sum_{k=1}^{K} P(x, h, s_k, t_k) \geq r \cdot \left[\sum_{k=1}^{K} \sum_{j=1}^{J(k)} (v_k^{j-1} - v_k^j) \right] - (e' + e'') - K$$

$$- \left[\sum_{k=1}^{K} J(k) \right] \cdot (f' + f'')$$

$$+ \left[\sum_{k=1}^{K} [J(k) - 1] \right] \cdot [W(x, h) - 2]$$

$$= r \cdot \sum_{k=1}^{K} (v_k^0 - v_k^{J(k)}) - (e' + e'') - K \cdot (f' + f'' + 1) \tag{140}$$

$$+ \left[\sum_{k=1}^{K} [J(k) - 1] \right] \cdot [W(x, h) - f' - f'' - 2]$$

Applying definitions (112) and (113), assumption (109), and (131) yields the desired result:

$$\sum_{k=1}^{K} P(x, h, s_k, t_k) \geq r \cdot \sum_{k=1}^{K} (v_k^0 - v_k^{J(k)}) - e - K \cdot f$$

$$= r \cdot \sum_{k=1}^{K} (t_k - s_k) - e - K \cdot f$$

This completes the proofs of inequality (111) and Lemma 3.

4.2.3.2 Lemma 4: Lower Bound on Throughput of Upstream Subpath

Suppose a system has been specified that satisfies the assumptions of Chapter 2 and uses round robin scheduling. Let x be some session. Let T_1 and T_2 be times satisfying $0 \leq T_1 < T_2 \leq \infty$. (Note that T_2 is permitted to be infinite.) Suppose there exist real numbers r, G_1 and G_2 such that the following inequality holds for every hop h of x in the range $0 \leq h \leq H(x)$, for any positive integer K, and for all times $s_1, t_1, s_2, t_2, \ldots, s_K, t_K$ satisfying $T_1 \leq s_1 \leq t_1 \leq s_2 \leq t_2 \leq \cdots \leq s_K \leq t_K < T_2$:

$$\sum_{k=1}^{K} C(x, h, s_k, t_k) \geq r \cdot \sum_{k=1}^{K} (t_k - s_k) - G_1 - K \cdot G_2 \tag{141}$$

Suppose that

$$[H(x) + 1] \cdot (G_2 + 1) \leq W(x, h) < \infty \quad \text{for } 1 \leq h \leq H(x) \tag{142}$$

It follows that property (143) holds for each hop h of x in the range $0 \leq h \leq H(x)$:

(143) If K is any positive integer,
and if $s_1, t_1, s_2, t_2, \ldots, s_K, t_K$ are any times
satisfying $T_1 \leq s_1 \leq t_1 \leq s_2 \leq t_2 \leq \cdots \leq s_K \leq t_K < T_2$,
and if $B(x, h+1, \tau) < W(x, h+1)$ for all τ in $\bigcup_{k=1}^{K} (s_k, t_k)$, then

$$\sum_{k=1}^{K} P(x, h, s_k, t_k) \geq r \cdot \sum_{k=1}^{K} (t_k - s_k) - (h + 1) \cdot G_1 - K \cdot (h + 1 \cdot G_2 + h$$

Proof of Lemma 4

The proof is by forward induction on h. The base case (i.e., $h = 0$) follows from assumption (141) and the fact that $P(x, 0, s_k , t_k) = C(x, 0, s_k , t_k)$ during intervals when $B(x, 1, \tau) < W(x, 1)$ (since buffer 0 is never empty). For the induction step, fix a hop h of x in the range $1 \leq h \leq H(x)$. Property (143) is assumed to hold for hop $h-1$, and it will be shown to hold for hop h. Let K be any positive integer, and let $s_1 , t_1 , s_2 , t_2 , \ldots , s_K , t_K$ be any times such that

$$T_1 \leq s_1 \leq t_1 \leq s_2 \leq t_2 \leq \cdots \leq s_K \leq t_K < T_2 \tag{144}$$

and

$$B(x, h+1, \tau) < W(x, h+1) \text{ for all } \tau \text{ in } \bigcup_{k=1}^{K} [s_k , t_k) \tag{145}$$

The goal is to show that

$$\sum_{k=1}^{K} P(x, h, s_k , t_k) \geq r \cdot \sum_{k=1}^{K} (t_k - s_k) - (h + 1) \cdot G_1 - K \cdot [(h + 1) \cdot G_2 + h] \tag{146}$$

To prove (146), Lemma 3 will be used, with:

$$e' = h \cdot G_1 \qquad e'' = G_1$$

$$f' = h \cdot G_2 + h - 1 \qquad f'' = G_2$$

First, note that the induction hypothesis can be rephrased in terms of time variables v_k^i and u_k^i as follows:

(147) If $J(1), J(2), \ldots, J(K)$ are any positive integers, and if
$v_1^{J(1)}, u_1^{J(1)-1}, v_1^{J(1)-1}, u_1^{J(1)-2}, \ldots, v_1^1, u_1^0,$
$v_2^{J(2)}, u_2^{J(2)-1}, v_2^{J(2)-1}, u_2^{J(2)-2}, \ldots, v_2^1, u_2^0, \ldots,$
$v_K^{J(K)}, u_K^{J(K)-1}, v_K^{J(K)-1}, u_K^{J(K)-2}, \ldots, v_K^1, u_K^0$
is any nondecreasing sequence of times in $[T_1, T_2)$,
and if $B(x, h, \tau) < W(x, h)$ for all τ in $\bigcup_{k=1}^{K} \bigcup_{j=1}^{J(k)} [v_k^j, u_k^{j-1})$, then

$$\sum_{k=1}^{K} \sum_{j=1}^{J(k)} P(x, h-1, v_k^j, u_k^{j-1}) \geq r \cdot \sum_{k=1}^{K} \sum_{j=1}^{J(k)} (u_k^{j-1} - v_k^j) - h \cdot G_1 - \left[\sum_{k=1}^{K} J(k) \right] \cdot (h \cdot G_2 + h - 1)$$

Using (147) and (144), it is straightforward to verify condition (107) of Lemma 3. Now condition (108) of Lemma 3 will be verified. As in (108), suppose, for $k = 1, 2, \ldots, K$, that $J(k)$ is some positive integer, that $u_k^{J(k)}, v_k^{J(k)}, u_k^{J(k)-1}, v_k^{J(k)-1}, \ldots, u_k^1, v_k^1$ are some times satisfying $s_k \leq u_k^{J(k)} \leq v_k^{J(k)} \leq u_k^{J(k)-1} \leq v_k^{J(k)-1} \leq \cdots \leq u_k^1 \leq v_k^1 \leq t_k$, and that $B(x, h, \tau) > 0$ for all τ in $\bigcup_{j=1}^{J(k)} [u_k^j, v_k^j)$. By (145), then, session x will accept every chance offered to it by the round robin scheduler at hop h during $\bigcup_{k=1}^{K} \bigcup_{j=1}^{J(k)} (u_k^j, v_k^j]$:

$$\sum_{k=1}^{K} \sum_{j=1}^{J(k)} P(x, h, u_k^j, v_k^j) = \sum_{k=1}^{K} \sum_{j=1}^{J(k)} C(x, h, u_k^j, v_k^j) \tag{148}$$

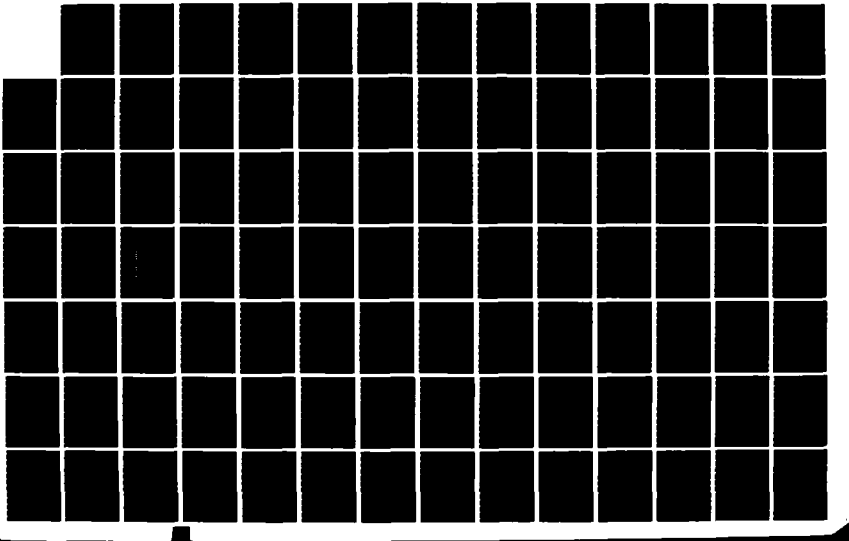
If follows from assumption (141) that

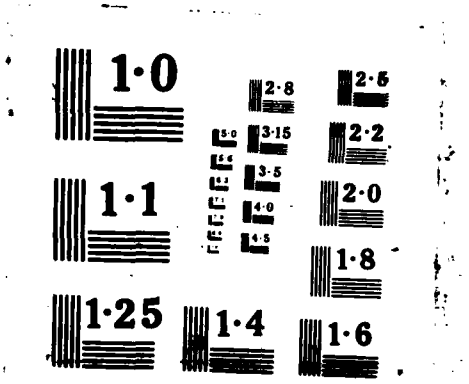
$$(149)\qquad \sum_{k=1}^{K}\sum_{j=1}^{J(k)} P(x,\, h,\, u_k^j\,,\, v_k^j) \;\geq\; r\cdot\sum_{k=1}^{K}\sum_{j=1}^{J(k)}(v_k^j - u_k^j) \;-\; G_1 \;-\; \left[\sum_{k=1}^{K} J(k)\right]\cdot G_2$$

This verifies condition (108). Condition (109) of Lemma 3 is satisfied by assumption (142). All the conditions of Lemma 3 have been verified. Conclusion (111) of Lemma 3 gives the desired result (146). This completes the proof of Lemma 4.

AD-A176 064 ROUND ROBIN SCHEDULING FOR FAIR FLOW CONTROL IN DATA COMMUNICATION NETWOR..(U) MASSACHUSETTS INST OF TECH CAMBRIDGE LAB FOR INFORMATION AND D.. E L HAHNE DEC 86 2/3

UNCLASSIFIED LIDS-TH-1631 N00014-84-K-0357 F/G 17/2 NL




1·0
1·1
1·25
1·4
1·6
1·8
2·0
2·2
2·5
2·8
3·15
3·5
4·0
4·5

4.2.3.3 Lemma 5: Lower Bound on Throughput of Downstream Subpath

Suppose a system has been specified that satisfies the assumptions of Chapter 2 and uses round robin scheduling. Let x be some session. Let T_1 and T_2 be times satisfying $0 \leq T_1 < T_2 \leq \infty$. (Note that T_2 is permitted to be infinite.) Suppose there exist real numbers r, G_1 and G_2 such that the following inequality holds for every hop h of x in the range $0 \leq h \leq H(x)$, for any positive integer K, and for all times $s_1, t_1, s_2, t_2, \ldots, s_K, t_K$ satisfying $T_1 \leq s_1 \leq t_1 \leq s_2 \leq t_2 \leq \cdots \leq s_K \leq t_K < T_2$:

$$(150) \qquad \sum_{k=1}^{K} C(x, h, s_k, t_k) \geq r \cdot \sum_{k=1}^{K} (t_k - s_k) - G_1 - K \cdot G_2$$

Suppose that

$$(151) \qquad [H(x) + 1] \cdot (G_2 + 1) \leq W(x, h) < \infty \quad \text{for } 1 \leq h \leq H(x)$$

It follows that property (152) holds for each hop h of x in the range $0 \leq h \leq H(x)$:

(152) If K is any positive integer,
and if $s_1, t_1, s_2, t_2, \ldots, s_K, t_K$ are any times
satisfying $T_1 \leq s_1 \leq t_1 \leq s_2 \leq t_2 \leq \cdots \leq s_K \leq t_K < T_2$,
and if $B(x, h, \tau) > 0$ for all τ in $\bigcup_{k=1}^{K} [s_k, t_k)$, then

$$\sum_{k=1}^{K} P(x, h, s_k, t_k) \geq r \cdot \sum_{k=1}^{K} (t_k - s_k) - [H(x) - h + 1] \cdot G_1 - K \cdot \Big[[H(x) - h + 1] \cdot G_2 + H(x) - h \Big]$$

Proof of Lemma 5

The proof is by backward induction on h. The base case (i.e., $h = H(x)$) follows from assumption (150) and the fact that $P(x, H(x), s_k , t_k) = C(x, H(x), s_k , t_k)$ during intervals when $B(x, H(x), \tau) > 0$ (since buffer $H(x)+1$ is never full). For the induction step, fix a hop h of x in the range

$$1 \leq h \leq H(x) \tag{153}$$

Property (152) is assumed to hold for hop h, and it will be shown to hold for hop $h-1$. Let K be any positive integer, and let $s_1 , t_1 , s_2 , t_2 , \ldots , s_K , t_K$ be any times such that

$$T_1 \leq s_1 \leq t_1 \leq s_2 \leq t_2 \leq \cdots \leq s_K \leq t_K < T_2 \tag{154}$$

and

$$B(x, h-1, \tau) > 0 \text{ for all } \tau \text{ in } \bigcup_{k=1}^{K} [s_k , t_k) \tag{155}$$

The goal is to show that

$$\sum_{k=1}^{K} P(x, h-1, s_k , t_k) \geq r \cdot \sum_{k=1}^{K} (t_k - s_k) - [H(x) - h + 2] \cdot G_1 - K \cdot \Big[[H(x) - h + 2] \cdot G_2 + H(x) - h + 1 \Big] \tag{156}$$

To prove (156), Lemma 3 will be used, with:

$$e' = G_1 \qquad e'' = [H(x) - h + 1]\cdot G_1$$

$$f' = G_2 \qquad f'' = [H(x) - h + 1]\cdot G_2 + H(x) - h$$

First, condition (107) of Lemma 3 will be verified. As in (107), suppose, for $k = 1, 2, \ldots, K$, that $J(k)$ is some positive integer, that $v_k^{J(k)}, u_k^{J(k)-1}, v_k^{J(k)-1}, u_k^{J(k)-2}, \ldots, v_k^1, u_k^0$ are some times satisfying $s_k \leq v_k^{J(k)} \leq u_k^{J(k)-1} \leq v_k^{J(k)-1} \leq u_k^{J(k)-2} \leq \cdots \leq v_k^1 \leq u_k^0 \leq t_k$, and that $B(x, h, \tau) < W(x, h)$ for all τ in $\bigcup_{j=1}^{J(k)} [v_k^j, u_k^{j-1})$. By (155), then, session x will accept every chance offered to it by the round robin scheduler at hop $h-1$ during $\bigcup_{k=1}^{K} \bigcup_{j=1}^{J(k)} (v_k^j, u_k^{j-1}]$:

$$\sum_{k=1}^{K} \sum_{j=1}^{J(k)} P(x, h-1, v_k^j, u_k^{j-1}) = \sum_{k=1}^{K} \sum_{j=1}^{J(k)} C(x, h-1, v_k^j, u_k^{j-1}) \tag{157}$$

If follows from assumption (150) that

$$\sum_{k=1}^{K} \sum_{j=1}^{J(k)} P(x, h-1, v_k^j, u_k^{j-1}) \geq r\cdot \sum_{k=1}^{K} \sum_{j=1}^{J(k)} (u_k^{j-1} - v_k^j) - G_1 - \left[\sum_{k=1}^{K} J(k)\right]\cdot G_2 \tag{158}$$

This verifies condition (107). Next, note that the induction hypothesis can be rephrased in terms of time variables u_k^j and v_k^j as follows:

(159) If $J(1), J(2), \ldots, J(K)$ are any positive integers, and if

$$u_1^{J(1)}, v_1^{J(1)}, u_1^{J(1)-1}, v_1^{J(1)-1}, \ldots, u_1^1, v_1^1,$$
$$u_2^{J(2)}, v_2^{J(2)}, u_2^{J(2)-1}, v_2^{J(2)-1}, \ldots, u_2^1, v_2^1, \ldots,$$
$$u_K^{J(K)}, v_K^{J(K)}, u_K^{J(K)-1}, v_K^{J(K)-1}, \ldots, u_K^1, v_K^1$$

is any nondecreasing sequence of times in $[T_1, T_2)$,

and if $B(x, h, \tau) > 0$ for all τ in $\bigcup_{k=1}^{K} \bigcup_{j=1}^{J(k)} [u_k^j, v_k^j)$, then

$$\sum_{k=1}^{K} \sum_{j=1}^{J(k)} P(x, h, u_k^j, v_k^j) \geq r \cdot \sum_{k=1}^{K} \sum_{j=1}^{J(k)} (v_k^j - u_k^j) - [H(x) - h + 1] \cdot G_1$$
$$- \left[\sum_{k=1}^{K} J(k) \right] \cdot \left[[H(x) - h + 1] \cdot G_2 + H(x) - h \right]$$

Using (159) and (154), it is straightforward to verify condition (108) of Lemma 3. Condition (109) of Lemma 3 is satisfied by assumptions (151) and (153). All the conditions of Lemma 3 have been verified. Conclusion (110) of Lemma 3 gives the desired result (156). This completes the proof of Lemma 5.

4.2.3.4 Lemma 6: Lower Bound on Throughput, given Lower Bound on Chances

Suppose a system has been specified that satisfies the assumptions of Chapter 2 and uses round robin scheduling. Let x be some session. Let T_1 and T_2 be times satisfying $0 \leq T_1 < T_2 \leq \infty$. (Note that T_2 is permitted to be infinite.) Suppose there exist real numbers r, G_1 and G_2 such that the following inequality holds for every hop h of x in the range $0 \leq h \leq H(x)$, for any positive integer K, and for all times $s_1, t_1, s_2, t_2, \ldots, s_K, t_K$ satisfying $T_1 \leq s_1 \leq t_1 \leq s_2 \leq t_2 \leq \cdots \leq s_K \leq t_K < T_2$:

$$(160) \qquad \sum_{k=1}^{K} C(x, h, s_k, t_k) \geq r \cdot \sum_{k=1}^{K} (t_k - s_k) - G_1 - K \cdot G_2$$

Suppose that

$$(161) \qquad [H(x) + 1] \cdot (G_2 + 1) \leq W(x, h) < \infty \qquad \text{for } 1 \leq h \leq H(x)$$

It follows that, for each hop h of x in the range $0 \leq h \leq H(x)$, for any positive integer K, and for any times $s_1, t_1, s_2, t_2, \ldots, s_K, t_K$ satisfying $T_1 \leq s_1 \leq t_1 \leq s_2 \leq t_2 \leq \cdots \leq s_K \leq t_K < T_2$:

$$(162) \quad \sum_{k=1}^{K} P(x, h, s_k, t_k)$$

$$\geq r \cdot \sum_{k=1}^{K} (t_k - s_k) - [H(x) + 1] \cdot G_1 - K \cdot \left[[H(x) + 1] \cdot G_2 + H(x) \right]$$

Proof of Lemma 6

Let h be any hop of x in the range $0 \leq h \leq H(x)$. Let K be any positive integer, and let $s_1, t_1, s_2, t_2, \ldots, s_K, t_K$ be any times satisfying

$$T_1 \leq s_1 \leq t_1 \leq s_2 \leq t_2 \leq \cdots \leq s_K \leq t_K < T_2 \tag{163}$$

If $h = 0$, then (162) follows directly from Lemma 5, since $B(x, 0, \tau) > 0$ for all times $\tau \geq 0$; so assume that $h \geq 1$. To prove (162), Lemma 3 will be used, with:

$$e' = h \cdot G_1 \qquad e'' = [H(x) - h + 1] \cdot G_1$$

$$f' = h \cdot G_2 + h - 1 \qquad f'' = [H(x) - h + 1] \cdot G_2 + H(x) - h$$

First, note that Lemma 4 can be applied to hop $h-1$ and rephrased in terms of time variables v_k^j and u_k^j to yield the following property:

(164) If $J(1), J(2), \ldots, J(K)$ are any positive integers, and if
$v_1^{J(1)}, u_1^{J(1)-1}, v_1^{J(1)-1}, u_1^{J(1)-2}, \ldots, v_1^1, u_1^0,$
$v_2^{J(2)}, u_2^{J(2)-1}, v_2^{J(2)-1}, u_2^{J(2)-2}, \ldots, v_2^1, u_2^0, \ldots,$
$v_K^{J(K)}, u_K^{J(K)-1}, v_K^{J(K)-1}, u_K^{J(K)-2}, \ldots, v_K^1, u_K^0$
is any nondecreasing sequence of times in $[T_1, T_2)$,
and if $B(x, h, \tau) < W(x, h)$ for all τ in $\bigcup_{k=1}^{K} \bigcup_{j=1}^{J(k)} [v_k^j, u_k^{j-1})$, then

$$\sum_{k=1}^{K} \sum_{j=1}^{J(k)} P(x, h-1, v_k^j, u_k^{j-1}) \geq r \cdot \sum_{k=1}^{K} \sum_{j=1}^{J(k)} (u_k^{j-1} - v_k^j) - h \cdot G_1 - \left[\sum_{k=1}^{K} J(k) \right] \cdot (h \cdot G_2 + h - 1)$$

Using (164) and (163), it is straightforward to verify condition (107) of Lemma 3. Next, note that Lemma 5 can rephrased in terms of time variables u_k^j and v_k^j to yield the following property:

(165) If $J(1), J(2), \ldots, J(K)$ are any positive integers, and if
$u_1^{J(1)}, v_1^{J(1)}, u_1^{J(1)-1}, v_1^{J(1)-1}, \ldots, u_1^1, v_1^1,$
$u_2^{J(2)}, v_2^{J(2)}, u_2^{J(2)-1}, v_2^{J(2)-1}, \ldots, u_2^1, v_2^1, \ldots,$
$u_K^{J(K)}, v_K^{J(K)}, u_K^{J(K)-1}, v_K^{J(K)-1}, \ldots, u_K^1, v_K^1$
is any nondecreasing sequence of times in $[T_1, T_2)$,
and if $B(x, h, \tau) > 0$ for all τ in $\bigcup_{k=1}^{K} \bigcup_{j=1}^{J(k)} [u_k^j, v_k^j)$, then

$$\sum_{k=1}^{K} \sum_{j=1}^{J(k)} P(x, h, u_k^j, v_k^j) \geq r \cdot \sum_{k=1}^{K} \sum_{j=1}^{J(k)} (v_k^j - u_k^j) - [H(x) - h + 1] \cdot G_1$$
$$- \left[\sum_{k=1}^{K} J(k) \right] \cdot \left[[H(x) - h + 1] \cdot G_2 + H(x) - h \right]$$

Using (165) and (163), it is straightforward to verify condition (108) of Lemma 3. Condition (109) of Lemma 3 is satisfied by assumption (161). All the conditions of Lemma 3 have been verified. Conclusion (111) of Lemma 3 gives the desired result (162). This completes the proof of Lemma 6.

4.2.4 Lemma 7: Upper Bound on Throughput, given Lower Bound on Throughput

Suppose a system has been specified that satisfies the assumptions of Chapter 2 and uses any scheduling discipline. Suppose each session has a well-defined demand rate Suppose that

$$W' < \infty \tag{167}$$

Let x be some session. Let h be some hop of x in the range $0 \leq h \leq H(x)$. Let K be a positive integer, and let $s_1, t_1, s_2, t_2, \ldots, s_K, t_K$ be times satisfying $0 \leq s_1 \leq t_1 \leq s_2 \leq t_2 \leq \cdots \leq s_K \leq t_K$. Let G_1 be a real number satisfying

$$\sum_{k=1}^{K-1} P(x, h, t_k, s_{k+1}) \geq R_F(I(x)) \cdot \sum_{k=1}^{K-1} (s_{k+1} - t_k) - G_1 \tag{168}$$

Let Δ be a non-negative real number such that

$$C(x, 0, s_1, t_K) \leq \lambda(x) \cdot (t_K - s_1) + \Delta \tag{169}$$

Let

$$G_2 \geq 0 \tag{170}$$

be a real number such that, for every session y with $I(y) \leq I(x)$ and for every link l used by y,

$$P'(y, l, s_1, t_K) \geq R_F(I(y)) \cdot (t_K - s_1) - G_2 \tag{171}$$

It follows that

$$(172)\quad \sum_{k=1}^{K} P(x, h, s_k, t_k)$$

$$\leq R_F(I(x))\cdot \sum_{k=1}^{K} (t_k - s_k) + (N-1)\cdot G_2 + G_1 + W'\cdot H + \Delta$$

Proof of Lemma 7

From assumption (168), it follows that

$$\sum_{k=1}^{K} P(x, h, s_k, t_k) = P(x, h, s_1, t_K) - \sum_{k=1}^{K-1} P(x, h, t_k, s_{k+1})$$

$$(173)\quad \leq P(x, h, s_1, t_K) - R_F(I(x))\cdot \sum_{k=1}^{K-1} (s_{k+1} - t_k) + G_1$$

Now the term $P(x, h, s_1, t_K)$ will be bounded. Recall that, by the properties of the max-min flow criterion (Section 4.1), every session has at least one bottleneck hop. Let h^* be any bottleneck hop of x, $0 \leq h^* \leq H(x)$, and consider the following claim:

$$(174)\quad P(x, h, s_1, t_K) \leq P(x, h^*, s_1, t_K) + W'\cdot H$$

If $h = h^*$, then (174) is obviously true. If $h > h^*$, then (174) is true because $P(x, h, s_1, t_K)$ can be no more than $P(x, h^*, s_1, t_K)$ plus the total number of packets present at time s_1 in all the buffers between hops h^* and h; there are at most H such buffers, each of capacity at most W'. If $h < h^*$, then (174) is true because $P(x, h, s_1, t_K)$ can be no more than $P(x, h^*, s_1, t_K)$ plus the total amount of spare capacity at time s_1 in all the

buffers between hops h and h^* ; there are at most H such buffers, each of capacity at most W' . This proves (174). Inequalities (173) and (174) show that

$$(175)\quad \sum_{k=1}^{K} P(x, h, s_k , t_k)$$

$$\leq P(x, h^*, s_1 , t_K) - R_F(I(x))\cdot \sum_{k=1}^{K-1} (s_{k+1} - t_k) + G_1 + W'\cdot H$$

Now $P(x, h^*, s_1 , t_K)$ will be analyzed, using the properties of bottleneck hops. There are two cases to consider. If $h^* = 0$, it follows from assumption (169) and definition (65) that

$$P(x, h^*, s_1 , t_K) = P(x, 0, s_1 , t_K)$$

$$\leq C(x, 0, s_1 , t_K)$$

$$\leq \lambda(x)\cdot(t_K - s_1) + \Delta$$

$$(176)\qquad = R_F(I(x))\cdot(t_K - s_1) + \Delta$$

If $1 \leq h^* \leq H(x)$, let l denote the link corresponding to hop h^*, and let Y denote the set of sessions $y \neq x$ that use l. Note that, by definition (64), $I(y) \leq I(x)$ for all sessions y in Y. Obviously, x can only use slots in $(s_1 , t_K]$ that are not used by sessions in Y:

$$P(x, h^*, s_1, t_K) = P'(x, l, s_1, t_K)$$

$$\leq (t_K - s_1) - \sum_{y \in Y} P'(y, l, s_1, t_K) \tag{177}$$

Applying assumptions (171) and (170) and definition (64) yields:

$$P(x, h^*, s_1, t_K) \leq (t_K - s_1) - \sum_{y \in Y} \left[R_F(I(y)) \cdot (t_K - s_1) - G_2 \right]$$

$$= \left[1 - \sum_{y \in Y} R_F(I(y)) \right] \cdot (t_K - s_1) + |Y| \cdot G_2$$

$$\leq \left[1 - \sum_{y \in Y} R_F(I(y)) \right] \cdot (t_K - s_1) + (N - 1) \cdot G_2$$

$$= R_F(I(x)) \cdot (t_K - s_1) + (N - 1) \cdot G_2 \tag{178}$$

Since Δ and G_2 are non-negative, inequalities (176) (for the case where $h^* = 0$) and (178) (for the case where $h^* > 0$) may be combined into a single inequality:

$$P(x, h^*, s_1, t_K) \leq R_F(I(x)) \cdot (t_K - s_1) + \Delta + (N - 1) \cdot G_2 \tag{179}$$

Substituting (179) into (175) gives the desired result (172):

$$\sum_{k=1}^{K} P(x, h, s_k, t_k) \leq R_F(I(x))\cdot(t_K - s_1) - R_F(I(x))\cdot\sum_{k=1}^{K-1}(s_{k+1} - t_k)$$

$$+ (N-1)\cdot G_2 + G_1 + W'\cdot H + \Delta$$

$$= R_F(I(x))\cdot\sum_{k=1}^{K}(t_k - s_k)$$

$$+ (N-1)\cdot G_2 + G_1 + W'\cdot H + \Delta$$

This completes the proof of Lemma 7.

4.3 Transient Analysis of Smooth Demand Case

This section contains a single result, Theorem 2. The theorem analyzes a system during an interval (T_1, T_2) of smooth demand. Specifically, it is assumed that there exists a constant Δ such that the demand of each session x over each subinterval $(s, t]$ of (T_1, T_2) is within Δ packets of the nominal amount $\lambda(x)\cdot(t-s)$. Most window sizes are assumed to be at least $3\cdot(H+1)^S\cdot N^{S-1}\cdot(\Delta+2)$. Theorem 2 concludes that the throughput of each session x at each hop over each subinterval $(s, t]$ of (T_1, T_2) is within $(H+1)^S\cdot N^{2S-1}\cdot(W'+3\Delta+4)$ packets of the fair amount $R_F(I(x))\cdot(t-s)$. Note that this unfairness bound increases with the maximum window size W'. This is not surprising, since the system should go through a transient period during which buffers upstream of bottleneck hops fill and buffers downstream of bottleneck hops drain. One would not expect to see fair flows until the buffers levels stabilize. Obviously, the transient can be more pronounced if the windows are larger.

Let us outline the proof of Theorem 2. The proof is by induction on the congestion index i of a session. The induction hypothesis gives upper and lower bounds on the throughput of each session with congestion index less than i. Recall that the theorem assumes upper and lower bounds on the demand of each session. The proof of the induction step has three parts. First, Lemma 2 uses the upper bound on throughput from the induction hypothesis plus the properties of round robin scheduling to deduce a lower bound on the number

of chances offered to each session with congestion index i at each link. Then, Lemma 6 uses this derived lower bound on chances, the given lower bound on demand, and the assumption of large windows to deduce a lower bound on the throughput of each session with congestion index i. Finally, Lemma 7 uses this derived lower bound on throughput, the lower bound on throughput from the induction hypothesis, the given upper bound on demand, and the properties of max-min fairness (viz., the existence of bottleneck hops) to deduce an upper bound on the throughput of each session with congestion index i. This preview should make the proof a little easier to follow.

4.3.1 Theorem 2: Throughput Bounds

Suppose a system has been specified that satisfies the assumptions of Chapter 2 and uses round robin scheduling. Suppose each session x has a well-defined, real demand rate $\lambda(x)$, $0 \leq \lambda(x) \leq 1$. Let T_1 and T_2 be times satisfying $0 \leq T_1 < T_2 \leq \infty$. (Note that T_2 is permitted to be infinite.) Suppose there exists a nonnegative real number Δ such that, for each session x and for all times s and t satisfying $T_1 \leq s \leq t < T_2$,

$$(180) \qquad | C(x, 0, s, t) - \lambda(x)\cdot(t - s) | \leq \Delta$$

Suppose that, for each session x,

$$(181) \quad 3\cdot(H+1)^S\cdot N^{S-1}\cdot(\Delta+2) \leq W(x, h) < \infty \quad \text{for } 1 \leq h \leq H(x)$$

It follows that, for each session x, for each hop h of x in the range $0 \leq h \leq H(x)$, and for all times s and t satisfying $T_1 \leq s \leq t < T_2$,

$$(182) \quad | P(x, h, s, t) - R_F(I(x))\cdot(t - s) | \leq (H+1)^S\cdot N^{2S-1}\cdot(W' + 3\Delta + 4)$$

Proof of Theorem 2

In order to show (182), properties (183) - (185) will be proved.

(183) For each session x, for each hop h of x in the range $0 \leq h \leq H(x)$, for any positive integer K, and for any times $s_1, t_1, s_2, t_2, \ldots, s_K, t_K$ satisfying $T_1 \leq s_1 \leq t_1 \leq s_2 \leq t_2 \leq \cdots \leq s_K \leq t_K < T_2$:

$$\sum_{k=1}^{K} C(x, h, s_k, t_k) \geq R_C(x, h)\cdot \sum_{k=1}^{K} (t_k - s_k) - D_{CL}(\Delta, I(x), K)$$

(184) For each session x, for each hop h of x in the range $0 \leq h \leq H(x)$, for any positive integer K, and for any times $s_1, t_1, s_2, t_2, \ldots, s_K, t_K$ satisfying $T_1 \leq s_1 \leq t_1 \leq s_2 \leq t_2 \leq \cdots \leq s_K \leq t_K < T_2$:

$$\sum_{k=1}^{K} P(x, h, s_k, t_k) \geq R_F(I(x)) \cdot \sum_{k=1}^{K} (t_k - s_k) - D_{PL}(\Delta, I(x), K)$$

(185) For each session x, for each hop h of x in the range $0 \leq h \leq H(x)$, for any positive integer K, and for any times $s_1, t_1, s_2, t_2, \ldots, s_K, t_K$ satisfying $T_1 \leq s_1 \leq t_1 \leq s_2 \leq t_2 \leq \cdots \leq s_K \leq t_K < T_2$:

$$\sum_{k=1}^{K} P(x, h, s_k, t_k) \leq R_F(I(x)) \cdot \sum_{k=1}^{K} (t_k - s_k) + D_{PU}(\Delta, I(x), K)$$

The proof is by induction on the congestion index $I(x)$ of the session x. Contrary to custom, the induction step will be proved before the base case is addressed. Fix a congestion index $i > 1$. The induction hypothesis asserts that (183), (184), and (185) hold for all sessions x with $I(x) < i$. It must be shown that (183), (184), and (185) hold for all sessions x with $I(x) = i$.

First consider (183). Let x be any session with $I(x) = i$. Let h be any hop of x in the range $0 \leq h \leq H(x)$. Let K be any positive integer, and let $s_1, t_1, s_2, t_2, \ldots, s_K, t_K$ be any times satisfying $T_1 \leq s_1 \leq t_1 \leq s_2 \leq t_2 \leq \cdots \leq s_K \leq t_K < T_2$. If hop $h = 0$, it follows from assumption (180), definition (68), and conclusion (90) of Lemma 1 that

$$\sum_{k=1}^{K} C(x, 0, s_k , t_k) \geq \lambda(x) \cdot \sum_{k=1}^{K} (t_k - s_k) - K \cdot \Delta$$

$$= R_C(x, 0) \cdot \sum_{k=1}^{K} (t_k - s_k) - K \cdot \Delta$$

$$\geq R_C(x, 0) \cdot \sum_{k=1}^{K} (t_k - s_k) - D_{CL}(\Delta, i, K) \tag{186}$$

If $1 \leq h \leq H(x)$, Lemma 2 will be used, with $G = D_{PU}(\Delta, i-1, K)$. By conclusion (84) of Lemma 1, condition (98) of Lemma 2 holds. By the induction hypothesis, (185) holds for all sessions with congestion index less than i. This fact, along with conclusion (88) of Lemma 1, verifies condition (99) of Lemma 2. From conclusion (100) of Lemma 2 and conclusion (92) of Lemma 1, it follows that

$$\sum_{k=1}^{K} C(x, h, s_k , t_k) \geq R_C(x, h) \cdot \sum_{k=1}^{K} (t_k - s_k) - (N-1) \cdot D_{PU}(\Delta, i-1, K) - K$$

$$\geq R_C(x, h) \cdot \sum_{k=1}^{K} (t_k - s_k) - D_{CL}(\Delta, i, K) \tag{187}$$

This completes the proof of (183) for the induction step.

Now (184) will be proved. Let x be any session with $I(x) = i$. Lemma 6 will be used, with $r = R_F(i)$, $G_1 = E_{CL}(\Delta, i)$ and $G_2 = F_{CL}(\Delta, i)$. To verify condition (160) of Lemma 6, let h be any hop of x in the range $0 \leq h \leq H(x)$, let K be any positive integer, and let s_1, t_1, s_2, t_2, . . . , s_K, t_K be any times satisfying

$T_1 \leq s_1 \leq t_1 \leq s_2 \leq t_2 \leq \cdots \leq s_K \leq t_K < T_2$. Recall that (183) was just proved for all sessions with congestion index i. Applying (183), (70), and definition (78) verifies condition (160):

$$\sum_{k=1}^{K} C(x, h, s_k, t_k) \geq R_C(x, h) \cdot \sum_{k=1}^{K} (t_k - s_k) - D_{CL}(\Delta, i, K)$$

$$\geq R_F(i) \cdot \sum_{k=1}^{K} (t_k - s_k) - D_{CL}(\Delta, i, K)$$

$$= R_F(i) \cdot \sum_{k=1}^{K} (t_k - s_k) - E_{CL}(\Delta, i) - K \cdot F_{CL}(\Delta, i) \tag{188}$$

Condition (161) of Lemma 6 holds because of assumption (181) and definition (74). Now conclusion (162) of Lemma 6, definition (78), and conclusion (94) of Lemma 1 can be applied to show that, for each hop h of x in the range $0 \leq h \leq H(x)$, for any positive integer K, and for any times s_1, t_1, s_2, t_2, ..., s_K, t_K satisfying $T_1 \leq s_1 \leq t_1 \leq s_2 \leq t_2 \leq \cdots \leq s_K \leq t_K < T_2$:

$$\sum_{k=1}^{K} P(x, h, s_k, t_k)$$

$$\geq R_F(i) \cdot \sum_{k=1}^{K} (t_k - s_k) - (H+1) \cdot E_{CL}(\Delta, i) - K \cdot [(H+1) \cdot F_{CL}(\Delta, i) + H]$$

$$= R_F(i) \cdot \sum_{k=1}^{K} (t_k - s_k) - (H+1) \cdot D_{CL}(\Delta, i, K) - K \cdot H$$

$$\geq R_F(i) \cdot \sum_{k=1}^{K} (t_k - s_k) - D_{PL}(\Delta, i, K)$$

This completes the proof of (184) for the induction step.

Next (185) will be proved. Let x be any session with $I(x) = i$. Let h be any hop of x in the range $0 \leq h \leq H(x)$. Let K be any positive integer, and let $s_1, t_1, s_2, t_2, \ldots, s_K, t_K$ be any times satisfying $T_1 \leq s_1 \leq t_1 \leq s_2 \leq t_2 \leq \cdots \leq s_K \leq t_K < T_2$. Lemma 7 will be used, with $G_1 = D_{PL}(\Delta, i, K-1)$ and $G_2 = D_{PL}(\Delta, i, 1)$. Condition (167) of Lemma 7 follows from assumption (181). If $K = 1$, condition (168) of Lemma 7 holds because of conclusion (82) of Lemma 1. If $K > 1$, condition (168) holds because (184) was just proved for all sessions with congestion index i. Condition (169) of Lemma 7 holds because of assumption (180). Condition (170) of Lemma 7 holds because of conclusion (82) of Lemma 1. By the induction hypothesis, (184) holds for all sessions with congestion index less than i. This fact, along with conclusion (86) of Lemma 1, verifies condition (171) of Lemma 7 for those sessions y with $I(y) < i$. For those sessions y with $I(y) = i$, condition (171) holds because (184) was just proved for all

sessions with congestion index i. From conclusion (172) of Lemma 7 and conclusion (97) of Lemma 1, it follows that

$$\sum_{k=1}^{K} P(x, h, s_k, t_k) \leq R_F(i) \cdot \sum_{k=1}^{K} (t_k - s_k) + (N-1) \cdot D_{PL}(\Delta, i, 1) + D_{PL}(\Delta, i, K-1) + W' \cdot H + \Delta$$

$$\leq R_F(i) \cdot \sum_{k=1}^{K} (t_k - s_k) + D_{PU}(\Delta, i, K) \tag{189}$$

This proves (185), completing the induction step.

The proof for the base case (i.e., $i = 1$) is identical to the induction step, considering the following point. In the induction step, the induction hypothesis was invoked to verify the assumptions of Lemmas 2 and 7 for sessions y with $I(y) < i$. For the base case, there are no such sessions y, so verifying these assumptions is trivial.

This completes the proofs of (183), (184), and (185). Conclusion (182) follows from results (184) - (185), definitions (71) - (80), and the fact that $I(x) \leq S$ for all sessions x.

This completes the proof of Theorem 2.

4.4 Steady-State Analysis of Smooth Demand Case

This section examines the steady-state behavior of systems with smooth demand. Specifically, it is assumed that there exists a constant Δ such that the demand of each session x over each interval $(s, t]$ is within Δ packets of the nominal amount $\lambda(x)\cdot(t-s)$. Most window sizes are assumed to be at least $3\cdot(H+1)^S\cdot N^{S-1}\cdot(\Delta+2)$. Corollary 1 of Theorem 2 concludes that the long-term average throughput $R_A(x)$ of each session x equals its fair rate $R_F(I(x))$. In other words, smooth demand and large windows are *sufficient* for throughput fairness. Example 3 shows that large windows are sometimes *necessary* as well, and that throughputs can be very unfair if the windows are too small. This example consists of eight links and $2N+1$ sessions, where N can be any even integer greater than four. The demand is perfectly smooth ($\Delta = 0$), but the window size $W(x, 2)$ for buffer 2 of a particular session x is less than $\frac{1}{2}N$. Because of this inadequate window size and an unfortunate choice of round robin rings and initial ring positions, the long-term average throughput of x is unfair by a factor of $\dfrac{N}{2\cdot W(x, 2)}$.

This section also presents a steady-state analog of Theorem 2. Theorem 3 states that there exists a time $T_{SS} \geq 0$ such that the throughput of each session x at each hop over each interval $(s, t]$ later than T_{SS} is within $(H+1)^S\cdot N^{S-1}\cdot(\Delta+2)$ packets of the fair amount $R_F(I(x))\cdot(t-s)$ and such that a similar lower bound applies to chances. (For concreteness, T_{SS} is

defined to be the earliest such time.) Note that the bound on throughput unfairness in steady state is tighter than the transient bound of Theorem 2. Moreover, the steady-state bound does not depend on the window sizes (except for the assumption that the windows are large enough).

The proof of Theorem 3 is similar to that of Theorem 2: the proof of Theorem 3 is also by induction on the congestion index, and the proof of the induction step also invokes Lemmas 2, 6 and 7 to generate, respectively, a lower bound on chances, a lower bound on throughput, and an upper bound on throughput for all sessions with a particular congestion index. The derived throughput bounds are of the form

$$(190) \qquad -f' \leq P(x, h, s, t) - R_F(I(x))\cdot(t-s) \leq f''$$

where f'' is a function of the maximum window size W' (and also of Δ, $I(x)$, H, and N), while f' does not depend on W', and $0 \leq f' \leq f''$. At this point Theorem 3 invokes Lemma 8 of Appendix A.1 to conclude that in steady state, i.e., for sufficiently large s and t,

$$(191) \qquad -f' \leq P(x, h, s, t) - R_F(I(x))\cdot(t-s) \leq f' + 1$$

In this way Theorem 3 derives bounds on the throughput unfairness in steady state that are tighter than the transient bounds of Theorem 2 and are independent of W'.

Included in this section are four corollaries of Theorem 3 about the steady-state buffer levels. In terms of the time T_{SS} (when the steady-state

throughput bounds take effect), let us define $m(x, h)$ and $M(x, h)$ for each buffer h of each session x such that $1 \leq h \leq H(x)+1$:

(192) $$m(x, h) = \min_{t \geq T_{ss}} B(x, h, t)$$

(193) $$M(x, h) = \max_{t \geq T_{ss}} B(x, h, t)$$

Since $B(x, 0, t) = \infty$ for all times $t \geq 0$, we also define

(194) $$m(x, 0) = M(x, 0) = \infty$$

Corollary 2 gives an upper bound on the range $M(x, h) - m(x, h)$ of a buffer level after time T_{SS} ; the bound does not depend on the window sizes (except for the assumption that the windows are large enough). Corollary 3 proves that, after time T_{SS} , buffers that are slightly upstream of bottleneck hops are sometimes full and are never empty, while buffers that are slightly downstream of bottleneck hops are sometimes empty and are never full. A bottleneck hop h^* of a session x in the range $0 \leq h^* \leq H(x)$ is called a *pure bottleneck* for x if there exists a time $T \geq 0$ such that all session x's chances at hop h^* after time T are successful:

(195) $$C(x, h^*, t-1, t) = P(x, h^*, t-1, t) \quad \text{for all times } t > T$$

In other words, the window mechanism does not impede the flow for a session at a pure bottleneck hop; packets and permits are always available whenever a chance for transmission arises. Corollary 4 concludes that every session has at least one pure bottleneck hop. Example 4 shows that impure bottlenecks do

exist. Corollary 5 asserts that a link at which sessions are bottlenecked is a pure bottleneck for all these sessions or for none of them.

4.4.1 Corollary 1: Fairness of Average Throughputs

Suppose a system has been specified that satisfies the assumptions of Chapter 2 and uses round robin scheduling. Suppose each session x has a well-defined, real demand rate $\lambda(x)$, $0 \leq \lambda(x) \leq 1$. Suppose there exists a nonnegative real number Δ such that, for each session x and for all times s and t satisfying $0 \leq s \leq t$,

$$| C(x, 0, s, t) - \lambda(x)\cdot(t - s) | \leq \Delta \tag{196}$$

Suppose that, for each session x,

$$3\cdot(H+1)^S\cdot N^{S-1}\cdot(\Delta+2) \leq W(x, h) < \infty \quad \text{for } 1 \leq h \leq H(x) \tag{197}$$

It follows that, for each session x, the long-term average throughput $R_A(x)$ exists and equals the fair rate $R_F(I(x))$.

Proof of Corollary 1

Note that the assumptions of Theorem 2 are satisfied, with $T_1 = 0$ and $T_2 = \infty$. Let x be any session. It follows from conclusion (182) of Theorem 2 that for all times $t > 0$,

$$| \frac{P(x, H(x), 0, t)}{t} - R_F(I(x)) | = \frac{| P(x, H(x), 0, t) - R_F(I(x))\cdot t |}{t}$$

$$\leq \frac{(H + 1)^S\cdot N^{2S-1}\cdot(W' + 3\Delta + 4)}{t} \tag{198}$$

It follows from definition (11) and (198) that $R_A(x)$ exists and that

$$R_A(x) = \lim_{t \to \infty} \frac{P(x, H(x), 0, t)}{t} = R_F(I(x)) \tag{199}$$

This completes the proof of Corollary 1.

4.4.2 Example 3: Unfairness with Small Windows

Consider a system that satisfies the assumptions of Chapter 2 and has the layout shown in Figure 8. The network contains links $l_{1,1}$, $l_{1,2}$, $l_{2,1}$, and $l_{2,2}$. (For each of these links, there is another link with opposite direction that is not shown in Figure 8 and is used only to return flow control permits.) For $j = 1, 2$, there are five sessions $y_{j,1}, y_{j,2}, \ldots, y_{j,5}$ that use $l_{j,1}$ followed by $l_{j,2}$, and there are five sessions $y_{j,6}, y_{j,7}, \ldots, y_{j,10}$ that use only $l_{j,1}$. There is also a session x that uses $l_{1,2}$ followed by $l_{2,2}$. Every session in the system has heavy demand; i.e.,

$$C(x, 0, t-1, t) = C(y_{j,k}, 0, t-1, t) = 1 \tag{200}$$

for $j = 1, 2$, for $k = 1, 2, \ldots, 10$, and for all times $t \geq 1$. The max-min fair rate for session x is $1/2$, while the other sessions deserve rates of $1/10$ each. The window size for each buffer $h \geq 1$ of each session is at least two but finite. In particular, $W(x, 2) = 2$, which is smaller than Theorem 3 requires. Table 1 shows the buffer levels at time 0. Round robin link scheduling is used. For $j = 1, 2$, the ring at $l_{j,1}$ is $y_{j,1}, y_{j,2}, \ldots, y_{j,10}$. The ring position of $l_{1,1}$ at time 0 is $y_{1,1}$, while the initial ring position of $l_{2,1}$ is $y_{2,6}$. For $j = 1, 2$, the ring at $l_{j,2}$ is $y_{j,1}, y_{j,2}, \ldots, y_{j,5}, x$. The ring position of $l_{1,2}$ at time 0 is x, while the initial ring position of $l_{2,2}$ is $y_{2,5}$.

This system is periodic, with a period of ten slots. Table 2 shows which

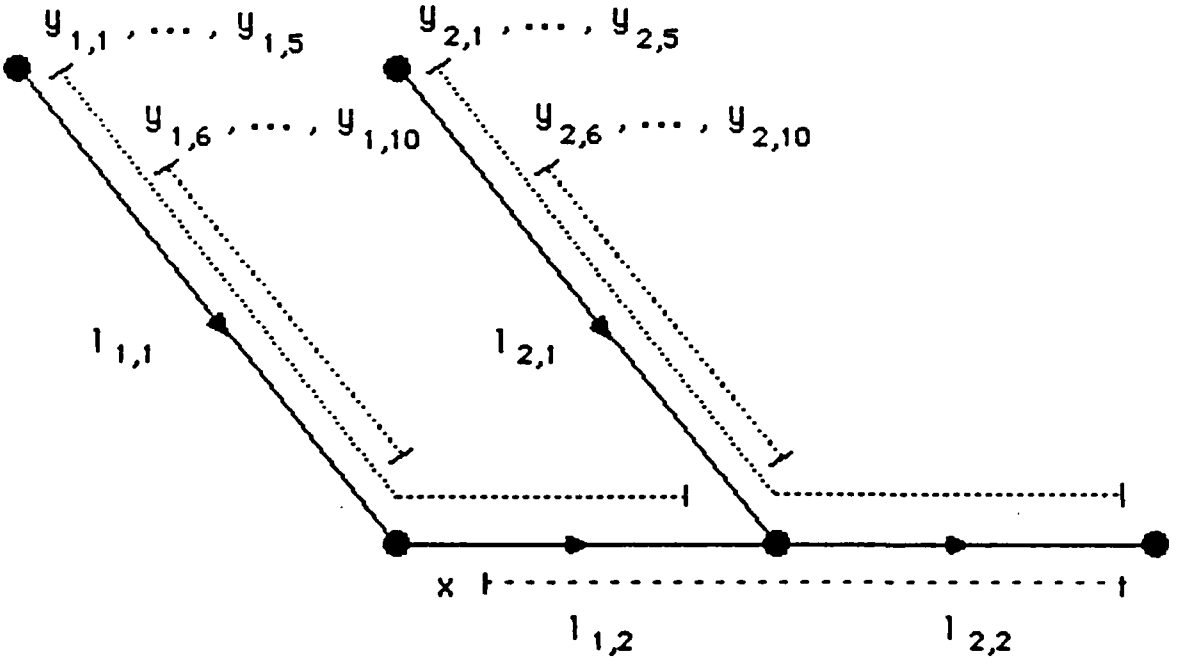

$y_{1,1}, \cdots, y_{1,5}$
$y_{2,1}, \cdots, y_{2,5}$
$y_{1,6}, \cdots, y_{1,10}$
$y_{2,6}, \cdots, y_{2,10}$
$l_{1,1}$
$l_{2,1}$
x
$l_{1,2}$
$l_{2,2}$

FIGURE 8

Session	Buffer Number			
	0	1	2	3
x	∞	$W(x, 1)$	$W(x, 2) = 2$	0
$y_{1,1}$	∞	$W(y_{1,1}, 1) - 1$	1	0
$y_{1,k}$, $2 \leq k \leq 5$	∞	$W(y_{1,k}, 1)$	0	0
$y_{1,k}$, $6 \leq k \leq 10$	∞	$W(y_{1,k}, 1)$	0	--
$y_{2,k}$, $1 \leq k \leq 4$	∞	$W(y_{2,k}, 1)$	0	0
$y_{2,5}$	∞	$W(y_{2,5}, 1)$	0	1
$y_{2,6}$	∞	$W(y_{2,6}, 1) - 1$	1	--
$y_{2,k}$, $7 \leq k \leq 10$	∞	$W(y_{2,k}, 1)$	0	--

TABLE 1. Initial Buffer Levels

Link	Slot Number									
	1	2	3	4	5	6	7	8	9	10
$l_{1,1}$	$y_{1,2}$	$y_{1,3}$	$y_{1,4}$	$y_{1,5}$	$y_{1,6}$	$y_{1,7}$	$y_{1,8}$	$y_{1,9}$	$y_{1,10}$	$y_{1,1}$
$l_{1,2}$	$y_{1,1}$	$y_{1,2}$	$y_{1,3}$	$y_{1,4}$	$y_{1,5}$	x	x	--	--	--
$l_{2,2}$	x	x	--	--	--	$y_{2,1}$	$y_{2,2}$	$y_{2,3}$	$y_{2,4}$	$y_{2,5}$
$l_{2,1}$	$y_{2,7}$	$y_{2,8}$	$y_{2,9}$	$y_{2,10}$	$y_{2,1}$	$y_{2,2}$	$y_{2,3}$	$y_{2,4}$	$y_{2,5}$	$y_{2,6}$

TABLE 2. Link Users over One Period

session uses each slot at each lir ..uring the interval (0, 10] . During the first half of this interval, session x ·nsmits no packets over link $l_{1,2}$ (because the link is busy serving other sessions) and only $W(x, 2) = 2$ packets over $l_{2,2}$ (because x runs out of packets in buffer 2). During the second half of the interval, x transmits no packets over $l_{2,2}$ (because the link is busy serving other sessions) and only $W(x, 2) = 2$ packets over $l_{1,2}$ (because x runs out of permits for buffer 2). The long-term average throughput of session x is $2/10$, well below its fair rate of $1/2$. The long-term average throughputs of the other sessions are fair. Three tenths of the capacities of $l_{1,2}$ and $l_{2,2}$ are wasted.

Using the same network, similar examples can be constructed that have different numbers of sessions and different window sizes. Let N be an even integer greater than four. For $j = 1, 2$, there are $½N$ sessions using $l_{j,1}$ and $l_{j,2}$ and $½N$ sessions using only $l_{j,1}$. As before, session x uses $l_{1,2}$ and $l_{2,2}$. Every session has heavy demand, so x has a fair rate of $1/2$, and the fair rate for every other session is $1/N$. The window size for each buffer of each session is at least two, and $2 \leq W(x, 2) < ½N$. The round robin rings and the initial conditions are such that the system has a period of N slots, x is served at $l_{2,2}$ during the first half of each period, and x is served at $l_{1,2}$ during the second half of each period. Consequently, x transmits only $W(x, 2)$ packets over each hop every N slots. In other words, the long-term average throughput of x is unfair by a factor of $\frac{N}{2 \cdot W(x, 2)}$. Moreover, the capacity x

loses at $l_{1,2}$ and $l_{2,2}$, viz., $\frac{1}{2} - \frac{W(x, 2)}{N}$, is not used by the other sessions -- it is wasted.

4.4.3 Theorem 3: Throughput Bounds in Steady State

Suppose a system has been specified that satisfies the assumptions of Chapter 2 and uses round robin scheduling. Suppose each session x has a well-defined, real demand rate $\lambda(x)$, $0 \leq \lambda(x) \leq 1$. Suppose there exists a nonnegative real number Δ such that, for each session x and for all times s and t satisfying $0 \leq s \leq t$,

$$(201) \qquad | C(x, 0, s, t) - \lambda(x) \cdot (t - s) | \leq \Delta$$

Suppose that, for each session x,

$$(202) \qquad 3 \cdot (H+1)^S \cdot N^{S-1} \cdot (\Delta+2) \leq W(x, h) < \infty \qquad \text{for } 1 \leq h \leq H(x)$$

It follows that there exists a time $T_{SS} \geq 0$ such that, for each session x, for each hop h of x in the range $0 \leq h \leq H(x)$, and for all times s and t satisfying $T_{SS} \leq s \leq t$,

$$(203) \qquad C(x, h, s, t) \geq R_C(x, h) \cdot (t - s) - (H + 1)^{S-1} \cdot N^{S-1} \cdot (\Delta + 2)$$

$$(204) \qquad | P(x, h, s, t) - R_F(I(x)) \cdot (t - s) | \leq (H + 1)^S \cdot N^{S-1} \cdot (\Delta + 2)$$

For concreteness, define T_{SS} to be the smallest nonnegative time with this property.

Proof of Theorem 3

In order to prove (203) and (204), it will be shown that there exist times

$T_{SS}(0)$, $T_{SS}(1)$, ..., $T_{SS}(I)$ such that $0 = T_{SS}(0) \leq T_{SS}(1) \leq \cdots \leq T_{SS}(I)$ and such that properties (205) - (208) hold.

(205) For each session x, for each hop h of x in the range $0 \leq h \leq H(x)$, and for all times s and t satisfying $T_{SS}(I(x)-1) \leq s \leq t$:
$C(x, h, s, t) \geq R_C(x, h)\cdot(t - s) - F_{CL}(\Delta, I(x))$

(206) For each session x, for each hop h of x in the range $0 \leq h \leq H(x)$, and for all times s and t satisfying $T_{SS}(I(x)-1) \leq s \leq t$:
$P(x, h, s, t) \geq R_F(I(x))\cdot(t - s) - F_{PL}(\Delta, I(x))$

(207) For each session x, for each hop h of x in the range $0 \leq h \leq H(x)$, and for all times s and t satisfying $T_{SS}(I(x)-1) \leq s \leq t$:
$P(x, h, s, t) \leq R_F(I(x))\cdot(t - s) + F''_{PU}(\Delta, I(x))$

(208) For each session x, for each hop h of x in the range $0 \leq h \leq H(x)$, and for all times s and t satisfying $T_{SS}(I(x)) \leq s \leq t$:
$P(x, h, s, t) \leq R_F(I(x))\cdot(t - s) + F_{PU}(\Delta, I(x))$

The proof is by induction on the congestion index $I(x)$ of the session x. Contrary to custom, the induction step will be proved before the base case is addressed. Fix a congestion index $i > 1$. The induction hypothesis asserts that there exist times $T_{SS}(0)$, $T_{SS}(1)$ ..., $T_{SS}(i-1)$ such that $0 = T_{SS}(0) \leq T_{SS}(1) \leq \cdots \leq T_{SS}(i-1)$ and such that properties (205) - (208) hold for all sessions x with $I(x) < i$. It must be shown that, for such a

time $T_{SS}(i-1)$, properties (205), (206), and (207) also hold for all sessions x with $I(x) = i$. It must also be shown that there exists a time $T_{SS}(i) \geq T_{SS}(i-1)$ such that property (208) holds for all sessions x with $I(x) = i$.

First consider (205). Let x be any session with $I(x) = i$. Let h be any hop of x in the range $0 \leq h \leq H(x)$. Let s and t be any times satisfying $T_{SS}(i-1) \leq s \leq t$. If hop $h = 0$, it follows from assumption (201), definition (68), and conclusion (89) of Lemma 1 that

$$\begin{aligned} C(x, 0, s, t) &\geq \lambda(x)\cdot(t - s) - \Delta \\ &= R_C(x, 0)\cdot(t - s) - \Delta \\ &\geq R_C(x, 0)\cdot(t - s) - F_{CL}(\Delta, i) \end{aligned} \tag{209}$$

If $1 \leq h \leq H(x)$, Lemma 2 will be used, with $K = 1$, $s_1 = s$, $t_1 = t$, and $G = F_{PU}(\Delta, i-1)$. By conclusion (83) of Lemma 1, condition (98) of Lemma 2 holds. By the induction hypothesis, (208) holds for all sessions with congestion index less than i. This fact, along with conclusion (87) of Lemma 1, verifies condition (99) of Lemma 2. From conclusion (100) of Lemma 2 and conclusion (91) of Lemma 1, it follows that

$$\begin{aligned} C(x, h, s, t) &\geq R_C(x, h)\cdot(t - s) - (N - 1)\cdot F_{PU}(\Delta, i-1) - 1 \\ &\geq R_C(x, h)\cdot(t - s) - F_{CL}(\Delta, i) \end{aligned} \tag{210}$$

This completes the proof of (205) for the induction step.

Now (206) will be proved. Let x be any session with $I(x) = i$. Lemma 6 will be used, with $T_1 = T_{SS}(i-1)$, $T_2 = \infty$, $r = R_F(i)$, $G_1 = 0$, and $G_2 = F_{CL}(\Delta, i)$. To verify condition (160) of Lemma 6, let h be any hop of x in the range $0 \leq h \leq H(x)$, let K be any positive integer, and let s_1, t_1, s_2, t_2, ..., s_K, t_K be any times satisfying $T_{SS}(i-1) \leq s_1 \leq t_1 \leq s_2 \leq t_2 \leq \cdots \leq s_K \leq t_K$. Recall that (205) was just proved for all sessions with congestion index i. It follows from (205) and (70) that

$$C(x, h, s_k, t_k) \geq R_C(x, h)\cdot(t_k - s_k) - F_{CL}(\Delta, i)$$

$$\geq R_F(i)\cdot(t_k - s_k) - F_{CL}(\Delta, i) \tag{211}$$

Summing (211) over k verifies condition (160). Condition (161) of Lemma 6 holds because of assumption (202) and definition (74). Now conclusion (162) of Lemma 6 and conclusion (93) of Lemma 1 can be applied to show that, for each hop h of x in the range $0 \leq h \leq H(x)$ and for all times s and t satisfying $T_{SS}(i-1) \leq s \leq t$:

$$P(x, h, s, t) \geq R_F(i)\cdot(t - s) - [(H + 1)\cdot F_{CL}(\Delta, i) + H]$$

$$\geq R_F(i)\cdot(t - s) - F_{PL}(\Delta, i) \tag{212}$$

This completes the proof of (206) for the induction step.

Next (207) will be proved. Let x be any session with $I(x) = i$. Let h be any hop of x in the range $0 \leq h \leq H(x)$. Let s and t be any times satisfying $T_{SS}(i-1) \leq s \leq t$. Lemma 7 will be used, with $K = 1$, $s_1 = s$, $t_1 = t$, $G_1 = 0$, and $G_2 = F_{PL}(\Delta, i)$. Condition (167) of Lemma 7 follows from assumption (202). Condition (168) of Lemma 7 is obviously true, since $K = 1$. Condition (169) of Lemma 7 holds because of assumption (201). Condition (170) of Lemma 7 holds because of conclusion (81) of Lemma 1. By the induction hypothesis, (206) holds for all sessions with congestion index less than i. This fact, along with conclusion (85) of Lemma 1, verifies condition (171) of Lemma 7 for those sessions y with $I(y) < i$. For those sessions y with $I(y) = i$, condition (171) holds because (206) was just proved for all sessions with congestion index i. From conclusion (172) of Lemma 7 and conclusion (96) of Lemma 1, it follows that

$$P(x, h, s, t) \leq R_F(i)\cdot(t - s) + (N - 1)\cdot F_{PL}(\Delta, i) + W'\cdot H + \Delta$$

$$\leq R_F(i)\cdot(t - s) + F''_{PU}(\Delta, i) \tag{213}$$

This completes the proof of (207) for the induction step.

Now (208) will be proved. Let x be any session with $I(x) = i$. Let h be any hop of x in the range $0 \leq h \leq H(x)$. Lemma 8 of Appendix A.1 will be used, with:

$$g(u) = P(x, h, u-1, u)$$

$$G(s, t) = P(x, h, s, t)$$

$$T = T_{SS}(i-1)$$

$$r = R_F(i)$$

$$f' = F_{PL}(\Delta, i)$$

$$f'' = F''_{PU}(\Delta, i)$$

$$\epsilon = 1$$

Condition (335) of Lemma 8 holds for all times s and t satisfying $T_{SS}(i-1) \leq s \leq t$ because (206) and (207) were just proved for all sessions with congestion index i. By Lemma 8, then, there exists a time $T_\epsilon(x, h) \geq T_{SS}(i-1)$ such that, for all times s and t satisfying $T_\epsilon(x, h) \leq s \leq t$,

(214) $$P(x, h, s, t) \leq R_F(i)\cdot(t - s) + F_{PL}(\Delta, i) + 1$$

Applying conclusion (95) of Lemma 1 yields:

(215) $$P(x, h, s, t) \leq R_F(i)\cdot(t - s) + F_{PU}(\Delta, i)$$

Define $T_{SS}(i)$ as follows:

(216) $$T_{SS}(i) = \max_{\substack{x:\ I(x)=i \\ h:\ 0\leq h\leq H(x)}} T_\epsilon(x, h)$$

This proves (208), completing the induction step.

Next the base case (i.e., $i = 1$) will be considered. Note that

$T_{SS}(i-1) = T_{SS}(0) = 0$ by definition. The proof for the base case is identical to the induction step, considering the following point. In the induction step, the induction hypothesis was invoked to verify the assumptions of Lemmas 2 and 7 for sessions y with $I(y) < i$. For the base case, there are no such sessions y, so verifying these assumptions is trivial.

This completes the proofs of (205) - (208). From (205) - (208), definitions (74) - (76), and the fact that $I(x) \leq S$ for all sessions x, it follows that (203) and (204) hold for each session x, for each hop h of x in the range $0 \leq h \leq H(x)$, and for all times s and t satisfying $T_{SS}(I) \leq s \leq t$.

This completes the proof of Theorem 3.

4.4.4 Corollary 2: Bound on Buffer Level Range

It follows from the assumptions of Theorem 3 that, for each session x,

$$M(x, h) - m(x, h) \leq 2 \cdot (H+1)^S \cdot N^{S-1} \cdot (\Delta+2) \quad \text{for } 1 \leq h \leq H(x) \tag{217}$$

Proof of Corollary 2

Choose any times $s \geq T_{SS}$ and $t \geq T_{SS}$ such that

$$B(x, h, s) = m(x, h) \tag{218}$$

$$B(x, h, t) = M(x, h) \tag{219}$$

Suppose that $s \leq t$. (The proof for $s > t$ is similar and will not be presented.) Applying (10) and conclusion (204) of Theorem 3 gives the desired result:

$$\begin{aligned} M(x, h) - m(x, h) &= B(x, h, t) - B(x, h, s) \\ &= P(x, h-1, s, t) - P(x, h, s, t) \\ &\leq \left[R_F(I(x)) \cdot (t - s) + (H + 1)^S \cdot N^{S-1} \cdot (\Delta + 2)\right] \\ &\quad - \left[R_F(I(x)) \cdot (t - s) - (H + 1)^S \cdot N^{S-1} \cdot (\Delta + 2)\right] \\ &= 2 \cdot (H + 1)^S \cdot N^{S-1} \cdot (\Delta + 2) \end{aligned}$$

This completes the proof of Corollary 2.

4.4.5 Corollary 3: Effect of Bottleneck Locations on Buffer Levels

Suppose that the assumptions of Theorem 3 hold. Let x be some session. Let J be the number of bottleneck hops for x, and let $h^*_1, h^*_2, \ldots, h^*_J$ denote the hop numbers of the bottlenecks of x, with

$$0 \leq h^*_1 < h^*_2 < \cdots < h^*_J \leq H(x)$$

Under the assumptions above, properties (220) and (221) hold.

(220) For each hop h of x in the range $0 \leq h \leq H(x)$, at least one of the following statements is true:

(a) h is a bottleneck hop for x

(b) $m(x, h) = 0$

(c) $M(x, h+1) = W(x, h+1)$

(221) For each buffer h of x, $0 \leq h \leq H(x)+1$, at least one of the following statements is true:

(a) $m(x, h) > 0$

(b) $M(x, h) < W(x, h)$

Furthermore, there exist buffers $h'_0, h'_1, \ldots, h'_J$ of x (called *crossover buffers*) with

(222)

$$0 = h'_0 \leq h^*_1 < h'_1 \leq h^*_2 < h'_2 \leq \cdots \leq h^*_{J-1} < h'_{J-1} \leq h^*_J < h'_J = H(x)+1$$

such that properties (223) and (224) hold.

(223) If $h'_j < h \leq h^*_{j+1}$ for some j, $0 \leq j \leq J-1$, then

(a) $M(x, h) = W(x, h)$

(b) $m(x, h) > 0$

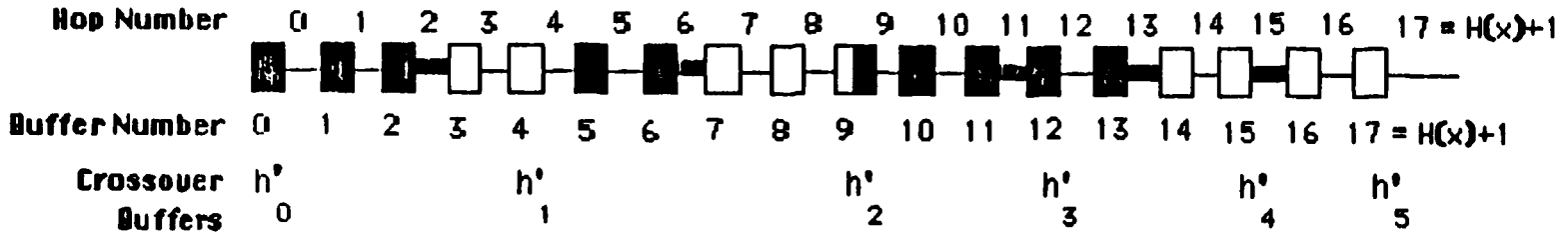
(224) If $h^*_j < h < h'_j$ for some j, $1 \leq j \leq J$, then

(a) $m(x, h) = 0$

(b) $M(x, h) < W(x, h)$

For the crossover buffers h'_j, $0 \leq j \leq J$, no claim stronger than (221) is made.

A combination of bottleneck locations and buffer levels that is consistent with properties (220) - (224) is shown in Figure 9. The figure shows the buffers and hops of a session x with a 16-link path and five bottleneck hops. Each buffer h is depicted as a square whose shading gives information about $m(x, h)$ and $M(x, h)$. The crossover buffers h'_j are also indicated. Hops are shown as lines between buffers. The heavier lines are the bottleneck hops. Examine the buffers between two successive bottleneck hops; note that the buffers upstream of the crossover buffer are sometimes empty and are never full (for times $t \geq T_{SS}$), while the buffers downstream of the crossover buffer are sometimes full and are never empty (for times $t \geq T_{SS}$).



means $m(x,h) > 0$ and $M(x,h) = W(x,h)$

means $m(x,h) = 0$ and $M(x,h) < W(x,h)$

means $m(x,h) > 0$ and $M(x,h) < W(x,h)$

FIGURE 9

Proof of Corollary 3

Property (220) will be proved by contradiction. Suppose that there is some non-bottleneck hop h of x in the range $0 \leq h \leq H(x)$ such that

$$m(x, h) > 0$$

$$M(x, h+1) < W(x, h+1)$$

This means that, for all times $t \geq T_{SS}$,

$$B(x, h, t) > 0$$

$$B(x, h+1, t) < W(x, h+1)$$

Therefore, session x will accept every chance offered to it by the round robin scheduler at hop h after time T_{SS} ; i.e., for all times $t > T_{SS}$,

(225) $$P(x, h, T_{SS}, t) = C(x, h, T_{SS}, t)$$

It follows from conclusion (203) of Theorem 3 that, for all times $t > T_{SS}$,

(226) $$P(x, h, T_{SS}, t) \geq R_C(x, h)\cdot(t - T_{SS}) - (H + 1)^{S-1}\cdot N^{S-1}\cdot(\Delta + 2)$$

Since h is not a bottleneck, (70) says that $R_C(x, h) > R_F(I(x))$. Therefore, (226) will violate conclusion (204) of Theorem 3 if t is large enough. This proves (220).

Next, property (221) will be proved. For $h = 0$, (G02a) is true. For $h = H(x)+1$, (G02b) is true. For the remaining buffers $h = 1, 2, \ldots, H(x)$, property (221) will be proved by contradiction. Suppose there is some buffer h of x in the range $1 \leq h \leq H(x)$ such that

(227) $$m(x, h) = 0$$

(228) $$M(x, h) = W(x, h)$$

It follows from assumption (202) of Theorem 3 that

$$M(x, h) - m(x, h) = W(x, h)$$

(229) $$\geq 3 \cdot (H + 1)^S \cdot N^{S-1} \cdot (\Delta + 2)$$

This contradicts conclusion (217) of Corollary 2, proving (221).

It was given that $h'_0 = 0$ and $h'_J = H(x)+1$. Let us now define crossover buffer h'_j for $1 \leq j \leq J-1$. If there is any buffer h in the range $h^*_j < h \leq h^*_{j+1}$ such that $M(x, h) < W(x, h)$, then let h'_j equal the largest such h; otherwise, let $h'_j = h^*_j + 1$.

Now, (223) will be proved. For $1 \leq j \leq J-1$, property (G03a) follows from the definition of h'_j, and (G03b) follows from (G03a) and (221). For $j = 0$, i.e., for

(230) $$0 = h'_0 < h \leq h^*_1$$

property (223) will be proved by forward induction on h. First consider the base case, $h = 1$. By (230), hop 0 is not a bottleneck hop, and since $m(x, 0) = \infty$, (220) implies that

(231) $$M(x, 1) = W(x, 1)$$

Together, (231) and (221) imply that

(232) $$m(x, 1) > 0$$

This proves the base case. The induction step is also proved by applying (220) followed by (221). This proves (223).

Next, (224) will be proved by backward induction on h. First consider the base case, viz.,

(233) $$h^*_j < h = h'_j - 1$$

For $j = 1, 2, \ldots, J-1$, the definition of h'_j and (233) imply that

(234) $$M(x, h'_j) < W(x, h'_j)$$

Note that (234) also holds for $j = J$, since $h'_J = H(x)+1$. Together, (220), (233) and (234) imply that

(235) $$m(x, h'_j - 1) = 0$$

Together, (221) and (235) imply that

(236) $$M(x, h'_j - 1) < W(x, h'_j - 1)$$

This proves the base case. The induction step is also proved by applying (220) followed by (221). This proves (224) and completes the proof of Corollary 3.

4.4.6 Corollary 4: Existence of Pure Bottlenecks

It follows from the assumptions of Theorem 3 that every session x has at least one pure bottleneck hop h^* in the range $0 \leq h^* \leq H(x)$.

Proof of Corollary 4

Let x be some session. It will be shown that x has at least one bottleneck hop h^* in the range $0 \leq h^* \leq H(x)$ with the following properties:

$$m(x, h^*) > 0$$

$$M(x, h^*+1) < W(x, h^*+1)$$

This means that session x will use all its chances at this hop h^* after time T_{SS}. In other words, this hop is a pure bottleneck for x.

The proof will be by contradiction. Suppose that, for each bottleneck hop h^* of x in the range $0 \leq h^* \leq H(x)$, at least one of the following statements is true:

$$m(x, h^*) = 0$$

$$M(x, h^*+1) = W(x, h^*+1)$$

This assumption, combined with conclusion (220) of Corollary 3, yields the following property.

(237) For each hop h of x in the range $0 \leq h \leq H(x)$, at least one of the following statements is true:

(a) $m(x, h) = 0$

(b) $M(x, h+1) = W(x, h+1)$

Now the following claim will be proved.

(238) For each buffer h of x in the range $0 \leq h \leq H(x)$,

$$m(x, h) > 0$$

The proof of (238) is by induction on h. The base case is true because $m(x, 0) = \infty$. The induction step is proved by applying (237) followed by conclusion (221) of Corollary 3.

For $h = H(x)$, (238) and (237) imply that

(239) $$M(x, H(x)+1) = W(x, H(x)+1)$$

This gives a contradiction, since buffer $H(x)+1$ is never full.

This completes the proof of Corollary 4.

4.4.7 Example 4: Existence of Impure Bottlenecks

Consider a system that satisfies the assumptions of Chapter 2 and has the layout shown in Figure 10. The network contains links $l_{1,1}$, $l_{1,2}$, $l_{2,1}$, $l_{2,2}$, and l_3. (For each of these links, there is another link with opposite direction that is not shown in Figure 10 and is used only to return flow control permits.) For $j = 1, 2$, there are two sessions $y_{j,1}$ and $y_{j,2}$ that use only link $l_{j,1}$, there are two sessions $y_{j,3}$ and $y_{j,4}$ that use links $l_{j,1}$ and $l_{j,2}$, and there is a session x_j that uses links $l_{j,2}$ and l_3. Every session in the system has heavy demand; i.e.,

$$C(x_j, 0, t-1, t) = C(y_{j,k}, 0, t-1, t) = 1 \tag{240}$$

for $j = 1, 2$, for $k = 1, 2, 3, 4$, and for all times $t \geq 1$. The max-min fair rates for x_1 and x_2 are ½ each, while the other sessions deserve rates of ¼ each. Notice that, for $j = 1, 2$, x_j is bottlenecked at both its links. The window size for each buffer $h \geq 1$ of each session is at least two but finite. Table 3 shows the buffer levels at time 0. Round robin link scheduling is used. For $j = 1, 2$, the ring at $l_{j,1}$ is $y_{j,1}$, $y_{j,2}$, $y_{j,3}$, $y_{j,4}$. The ring position of $l_{1,1}$ at time 0 is $y_{1,4}$, while the initial ring position of $l_{2,1}$ is $y_{2,2}$. For $j = 1, 2$, the ring at $l_{j,2}$ is $y_{j,3}$, $y_{j,4}$, x_j. The ring position of $l_{1,2}$ at time 0 is $y_{1,3}$, while the initial ring position of $l_{2,2}$ is x_2. The initial ring position of l_3 is x_1.

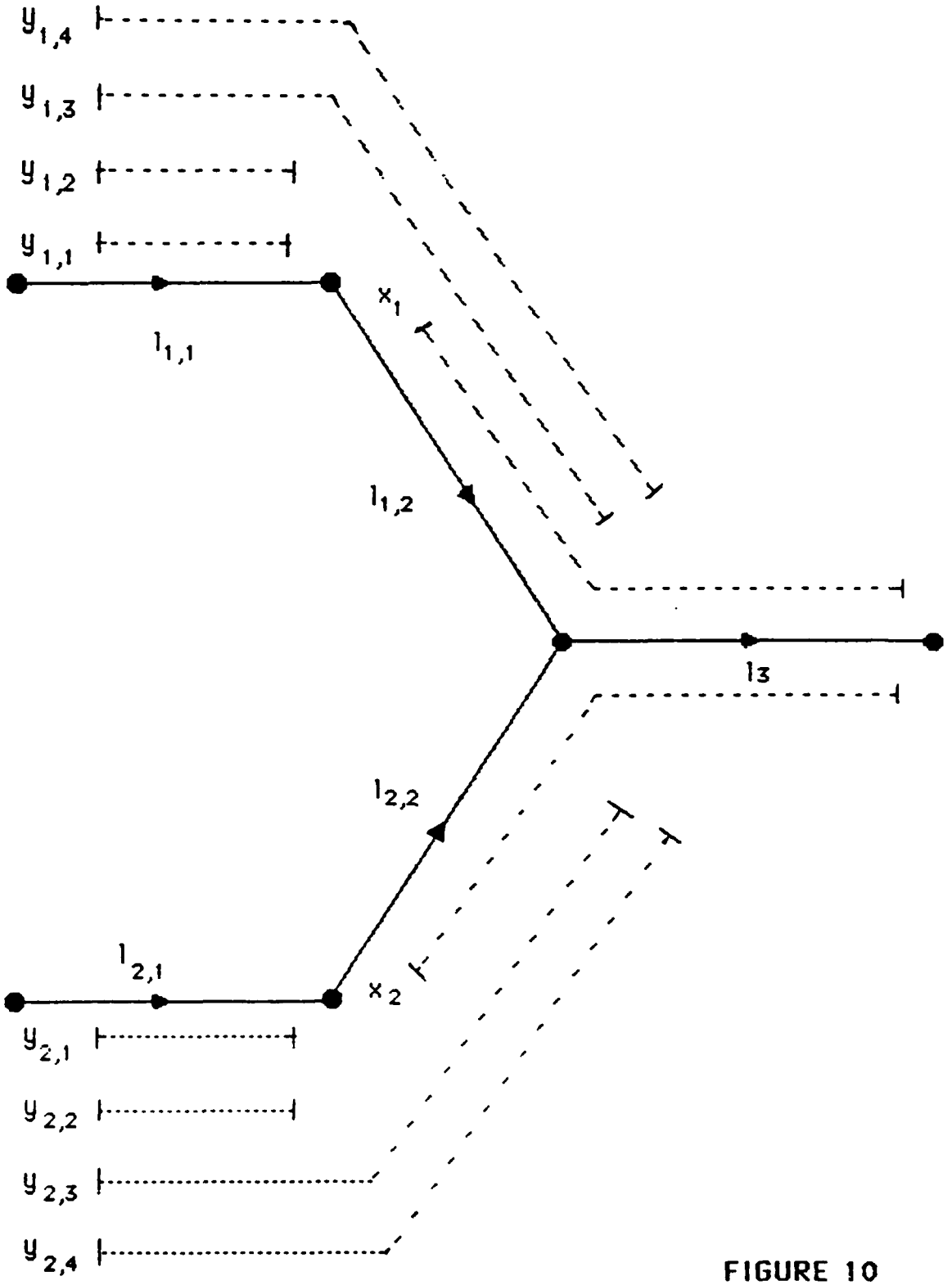


FIGURE 10

Session	Buffer Number			
	0	1	2	3
$y_{1,1}$	∞	$W(y_{1,1}, 1)$	0	--
$y_{1,2}$	∞	$W(y_{1,2}, 1)$	0	--
$y_{1,3}$	∞	$W(y_{1,3}, 1)$	0	1
$y_{1,4}$	∞	$W(y_{1,4}, 1) - 1$	1	0
x_1	∞	$W(x_1, 1)$	0	1
$y_{2,1}$	∞	$W(y_{2,1}, 1)$	0	--
$y_{2,2}$	∞	$W(y_{2,2}, 1) - 1$	1	--
$y_{2,3}$	∞	$W(y_{2,3}, 1)$	0	0
$y_{2,4}$	∞	$W(y_{2,4}, 1)$	0	0
x_2	∞	$W(x_2, 1) - 1$	1	0

TABLE 3. Initial Buffer Levels

This system is periodic, with a period of four slots. Table 4 shows which session uses each slot at each link during the interval $(0, 4]$. While the average session throughputs over every system period are max-min fair, sessions x_1 and x_2 each decline one chance to use link l_3 in every system period, because of a lack of packets. In other words, l_3 is an impure bottleneck for x_1 and x_2. This does not violate Corollary 4, however, because x_1 and x_2 have pure bottlenecks $l_{1,2}$ and $l_{2,2}$, respectively.

Link	Slot Number			
	1	2	3	4
$l_{1,1}$	$y_{1,1}$	$y_{1,2}$	$y_{1,3}$	$y_{1,4}$
$l_{1,2}$	$y_{1,4}$	x_1	x_1	$y_{1,3}$
l_3	x_2	x_2	x_1	x_1
$l_{2,2}$	x_2	$y_{2,3}$	$y_{2,4}$	x_2
$l_{2,1}$	$y_{2,3}$	$y_{2,4}$	$y_{2,1}$	$y_{2,2}$

TABLE 4. Link Users over One Period

4.4.8 Corollary 5: A Property of Pure Bottlenecks

Given the assumptions of Theorem 3, any link that is a pure bottleneck for some session must be a pure bottleneck for every session bottlenecked there.

Proof of Corollary 5

Let K be any integer satisfying

$$K \geq 2 \cdot (H+1)^S \cdot N^{S-1} \cdot (\Delta + 2) + 2 \tag{241}$$

The proof of Corollary 5 will be by contradiction. Let l be a bottleneck link for sessions x and y. Suppose that l is a pure bottleneck for x but not for y. In other words, there is a time after which session x accepts every chance offered to it by the round robin scheduler at link l; session y, on the other hand, declines infinitely many chances to use link l. Therefore, there exist times s and t such that $T_{SS} \leq s \leq t$ and such that x accepts every chance at l during $(s, t]$:

$$P'(x, l, s, t) = C'(x, l, s, t) \tag{242}$$

and such that y declines K chances at l during $(s, t]$:

$$P'(y, l, s, t) = C'(y, l, s, t) - K \tag{243}$$

By the operating rules of the round robin scheduler at l, x must receive almost as many chances as y during $(s, t]$; specifically:

$$C'(x, l, s, t) \geq C'(y, l, s, t) - 1 \tag{244}$$

Combine (242), (244), (243), and (241):

$$\begin{aligned} P'(x, l, s, t) &= C'(x, l, s, t) \\ &\geq C'(y, l, s, t) - 1 \\ &= P'(y, l, s, t) + K - 1 \end{aligned}$$

$$\geq P'(y, l, s, t) + 2 \cdot (H + 1)^S \cdot N^{S-1} \cdot (\Delta + 2) + 1 \tag{245}$$

Applying conclusion (204) of Theorem 3 to (245) yields:

$$P'(x, l, s, t) \geq R_F(I(y)) \cdot (t - s) + (H + 1)^S \cdot N^{S-1} \cdot (\Delta + 2) + 1 \tag{246}$$

Since x and y are bottlenecked at the same link, it follows from definition (64) that $I(x) = I(y)$. Substitute this into (246):

$$P'(x, l, s, t) \geq R_F(I(x)) \cdot (t - s) + (H + 1)^S \cdot N^{S-1} \cdot (\Delta + 2) + 1 \tag{247}$$

This contradicts conclusion (204) of Theorem 3, completing the proof of Corollary 5.

4.5 Steady-State Analysis of Bursty Demand Case

This section studies the long-term average session throughputs when the sessions have independent Bernoulli demand processes. Suppose such a system has been fully specified, including its initial state. For future reference, define $\alpha = \frac{\min_{x,h:1\le h\le H(x)} W(x,h)}{\max_{x,h:1\le h\le H(x)} W(x,h)}$. For each buffer h of each session x such that $1 \le h \le H(x)$, suppose that the window size $W(x,h)$ is at least $12\cdot(H+1)^S\cdot N^{S-1}$ but finite. Such a system can be modeled as a finite Markov chain [15] in which each state represents one combination of buffer levels and round robin ring positions. If the demand rate $\lambda(x)$ of each session x is strictly less than one, then the Markov chain has a single closed, communicating class of states. † However, if $\lambda(x) = 1$ for even one session x, then multiple classes are possible. Even from the given initial state, it may be possible to reach more than one of these classes. With probability one, the system will eventually enter one of these classes, after which, of course, it

† Suppose that every session's demand rate is strictly less than one. Number the sessions $x_1, x_2, \ldots, x_S$, put the system in an arbitrary state, and consider the following sequence of events. First, every session's demand is zero for long enough that every buffer $h \ge 1$ of every session empties. Then session x_1 transmits a single packet through the network, thereby setting the round robin ring positions to x_1 at each link of its path. After this packet has left buffer $H(x_1)+1$, session x_2 transmits one packet over its entire path, then session x_3 has its turn, etc. Since the resulting state is reachable from all states, the Markov chain has a single closed, communicating class of states.

cannot leave that class. For each session x and each closed, communicating class ξ, there exists a real number $r(x, \xi)$ with the following property: *given* that the system eventually ends up in class ξ, the long-term average throughput $R_A(x)$ of session x equals $r(x, \xi)$ with probability one. † This means that, given the initial state η, $R_A(x)$ is a random variable; $R_A(x)$ takes on value $r(x, \xi)$ according to the probability that the system enters class ξ from state η. With probability one, $R_A(x) = r(x, \xi)$ for *some* class ξ. Theorem 4 concludes that, for each session x and each class ξ, $r(x, \xi)$ differs from the fair rate $R_F(I(x))$ by no more than $\frac{74S\cdot(H+1)^{2S}\cdot N^{2S-1}}{\alpha\cdot(W')^{0.5}}$. In other words, $R_A(x)$ is within $\frac{74S\cdot(H+1)^{2S}\cdot N^{2S-1}}{\alpha\cdot(W')^{0.5}}$ of $R_F(I(x))$ for each session x, with probability one. This means that the session throughput rates can be made arbitrarily close to the fair rates by choosing window sizes that are of the same order of magnitude and are sufficiently large. (Example 5 demonstrates

---

† Let θ be some state in some closed, communicating class ξ of the Markov chain defined above, and let $\beta(\theta)$ denote its mean recurrence time. (Note that $1 \leq \beta(\theta) < \infty$, since θ is recurrent and the Markov chain is finite [15].) By applying the strong law of large numbers to the recurrence times of θ, it is easy to prove the following claim: given that the system enters class ξ, the long-term average number of visits to state θ per unit time is $1/\beta(\theta)$ with probability one. For each session x, let $\Theta(x, \xi)$ denote the set of states of ξ in which x has just transmitted a packet over hop $H(x)$ (i.e., in which buffer $H(x)+1$ contains a packet), and let $r(x, \xi) = \sum_{\theta \in \Theta(x,\xi)} 1/\beta(\theta)$. Given that the system enters class ξ, the long-term average throughput $R_A(x)$ of session x equals $r(x, \xi)$ with probability one.

why perfectly fair rates cannot be achieved, in general, with finite window sizes. †)

The proof of Theorem 4 is structured as follows. Time is divided into intervals of fixed length. Theorem 2 is used to bound the session throughputs during those intervals in which the demands of the sessions are fairly smooth. Lemma 9 of Appendix A.2 is used to bound the frequency of such intervals. Together, these results show that the session throughputs are nearly fair most of the time.

† Of course, infinite windows are not the solution: if unbounded queues build up in some buffers, then cross-network delay is also unbounded; moreover, a session's throughput can be (wastefully) higher on hops upstream of such buffers than on hops downstream.

4.5.1 Theorem 4: Approximate Fairness of Average Throughputs

Suppose a system has been specified that satisfies the assumptions of Chapter 2 and uses round robin scheduling. Suppose that the demands of the sessions are independent. For each session x, suppose chances at hop 0 form a Bernoulli process with rate $\lambda(x)$, $0 \leq \lambda(x) \leq 1$. Suppose that, for each session x,

$$(248) \qquad 12 \cdot (H+1)^S \cdot N^{S-1} \leq W(x,h) < \infty \quad \text{for } 1 \leq h \leq H(x)$$

Define a real number α as follows:

$$(249) \qquad \alpha = \frac{\min\limits_{x,h:\, 1 \leq h \leq H(x)} W(x,h)}{\max\limits_{x,h:\, 1 \leq h \leq H(x)} W(x,h)} = \frac{\min\limits_{x,h:\, 1 \leq h \leq H(x)} W(x,h)}{W'}$$

It follows that, with probability one, the long-term average throughput $R_A(x)$ exists for each session x and

$$(250) \qquad | R_A(x) - R_F(I(x)) | \leq \frac{74S \cdot (H+1)^{2S} \cdot N^{2S-1}}{\alpha \cdot (W')^{0.5}}$$

Proof of Theorem 4

Define a real number Δ as follows.

$$(251) \qquad \Delta = \frac{\alpha \cdot W'}{6 \cdot (H+1)^S \cdot N^{S-1}}$$

Since α is at most one and H, S and N are at least one, it follows from (251) and (248) that

$$W' + 3\Delta + 4 \leq W' + \frac{W'}{4} + \frac{W'}{6} \leq 2W' \tag{252}$$

It also follows from (251), (249) and (248) that $\Delta \geq 2$ and hence

$$\begin{aligned} 3\cdot(H+1)^S\cdot N^{S-1}\cdot(\Delta+2) &\leq 6\cdot(H+1)^S\cdot N^{S-1}\cdot\Delta \\ &= \alpha\cdot W' \\ &= \min_{x,h:\, 1\leq h\leq H(x)} W(x,h) \end{aligned} \tag{253}$$

Therefore, for each session x,

$$3\cdot(H+1)^S\cdot N^{S-1}\cdot(\Delta+2) \leq W(x,h) < \infty \quad \text{for } 1 \leq h \leq H(x) \tag{254}$$

These inequalities will prove useful later.

Now we proceed with the proof. Let τ be some integer in the range

$$\alpha\cdot(W')^{1.5} \leq \tau \leq 2\alpha\cdot(W')^{1.5} \tag{255}$$

Divide the time interval $(0, \infty)$ into non-overlapping subintervals of length τ slots: $(0, \tau]$, $(\tau, 2\tau]$, ... Label each subinterval "good" or "bad" according to the smoothness of the sessions' demands during that subinterval; a subinterval $((k-1)\cdot\tau, k\cdot\tau]$ is "good for session x " if

$$| C(x, 0, s, t) - \lambda(x)\cdot(t - s) | \leq \Delta \tag{256}$$

for all times s and t satisfying $(k-1)\cdot\tau \leq s < t \leq k\cdot\tau$. A subinterval is "bad for session x " if it is not good for x. Let $\pi(x)$ be the probability that a given subinterval is bad for session x. Lemma 9 of Appendix A.2 can be

applied, with $G(s, t) = C(x, 0, s, t)$, $\mu = \lambda(x)$, and $T = (k-1)\cdot\tau$, to conclude that

$$\pi(x) \leq \frac{\tau}{\Delta^2} \tag{257}$$

for all sessions x. A subinterval that is good for all sessions is simply called "good," while a subinterval that is bad for at least one session is called "bad." Let π be the probability that a given subinterval is bad. By (257),

$$\pi \leq \sum_x \pi(x)$$

$$\leq \sum_x \frac{\tau}{\Delta^2}$$

$$= \frac{S\cdot\tau}{\Delta^2} \tag{258}$$

For every positive integer K, let q_K denote the number of bad subintervals among $(0, \tau]$, $(\tau, 2\tau]$, ... , $((K-1)\cdot\tau, K\cdot\tau]$. Since demands during different subintervals are independent and identically distributed, the strong law of large numbers [3] can be applied to conclude that, with probability one,

$$\frac{q_K}{K} \xrightarrow[K\to\infty]{} \pi \tag{259}$$

Let Ω' be the set of sample paths for which (259) holds. As just mentioned,

$$PROB\{\Omega'\} = 1 \tag{260}$$

Let Ω'' be the set of sample paths for which the long-term average throughput $R_A(x)$ of every session x exists. As explained in the introduction to Section 4.5,

$$PROB\ \{\ \Omega''\ \}\ =\ 1 \tag{261}$$

It follows from (260) and (261) that

$$PROB\ \{\ \Omega' \cap \Omega''\ \}\ =\ 1 \tag{262}$$

It suffices to prove (250) for all sample paths in $\Omega' \cap \Omega''$ and all sessions x.

Let us restrict our attention to a particular sample path ω in $\Omega' \cap \Omega''$. Let x be any session. Let k be any positive integer. The throughput of x during subinterval $((k-1)\cdot\tau,\ k\cdot\tau\,]$ will now be analyzed. Since

$$0\ \leq\ P(x,\ H(x),\ (k-1)\cdot\tau,\ k\cdot\tau)\ \leq\ \tau \tag{263}$$

and

$$0\ \leq\ R_F(I(x))\ \leq\ 1 \tag{264}$$

it follows that

$$|\ P(x,\ H(x),\ (k-1)\cdot\tau,\ k\cdot\tau)\ -\ R_F(I(x))\cdot\tau\ |\ \leq\ \tau \tag{265}$$

Bound (265) will be used only for the bad subintervals. If $((k-1)\cdot\tau,\ k\cdot\tau\,]$ is a good subinterval, nicer throughput bounds can be obtained by using Theorem 2, with $T_1 = (k-1)\cdot\tau$ and $T_2 = k\cdot\tau + 1$. Conditions (180) and (181) of the theorem are satisfied by (256) and (254). Applying conclusion (182) of Theorem 2 and (252) yields:

$$| P(x, H(x), (k-1)\cdot\tau, k\cdot\tau) - R_F(I(x))\cdot\tau | \leq (H+1)^S \cdot N^{2S-1} \cdot (W' + 3\Delta + 4)$$

$$\leq 2W' \cdot (H+1)^S \cdot N^{2S-1} \tag{266}$$

Now, the throughput bounds (265) and (266) for the bad and good subintervals can be added together in the correct proportions in order to bound the throughput of x over longer intervals. For any positive integer K,

$$\left| \frac{P(x, H(x), 0, K\cdot\tau)}{K\cdot\tau} - R_F(I(x)) \right|$$

$$= \frac{\left| P(x, H(x), 0, K\cdot\tau) - R_F(I(x))\cdot K\cdot\tau \right|}{K\cdot\tau}$$

$$= \frac{\left| \sum_{k=1}^{K} [P(x, H(x), (k-1)\cdot\tau, k\cdot\tau) - R_F(I(x))\cdot\tau] \right|}{K\cdot\tau}$$

$$\leq \frac{\sum_{k=1}^{K} \left| P(x, H(x), (k-1)\cdot\tau, k\cdot\tau) - R_F(I(x))\cdot\tau \right|}{K\cdot\tau}$$

$$\leq \frac{q_K\cdot\tau + (K - q_K)\cdot[2W'\cdot(H+1)^S\cdot N^{2S-1}]}{K\cdot\tau}$$

$$\leq \frac{q_K\cdot\tau + K\cdot[2W'\cdot(H+1)^S\cdot N^{2S-1}]}{K\cdot\tau}$$

(267)
$$= \frac{q_K}{K} + \frac{2W'\cdot(H+1)^S\cdot N^{2S-1}}{\tau}$$

Since the sample path ω belongs to Ω'', the long-term average throughput $R_A(x)$ exists, and since ω belongs to Ω', (259) applies. Therefore, by definition (11), (267), and (259),

$$| R_A(x) - R_F(I(x)) | = | \lim_{t \to \infty} \frac{P(x, H(x), 0, t)}{t} - R_F(I(x)) |$$

$$= | \lim_{K \to \infty} \frac{P(x, H(x), 0, K \cdot \tau)}{K \cdot \tau} - R_F(I(x)) |$$

$$\leq \lim_{K \to \infty} \frac{q_K}{K} + \frac{2W' \cdot (H+1)^S \cdot N^{2S-1}}{\tau}$$

$$= \pi + \frac{2W' \cdot (H+1)^S \cdot N^{2S-1}}{\tau} \tag{268}$$

Applying (258), (255), and (251) to (268) gives the desired result (250):

$$| R_A(x) - R_F(I(x)) | \leq \frac{S \cdot \tau}{\Delta^2} + \frac{2W' \cdot (H+1)^S \cdot N^{2S-1}}{\tau}$$

$$\leq \frac{72S \cdot (H+1)^{2S} \cdot N^{2S-2}}{\alpha \cdot (W')^{0.5}} + \frac{2 \cdot (H+1)^S \cdot N^{2S-1}}{\alpha \cdot (W')^{0.5}}$$

$$\leq \frac{74S \cdot (H+1)^{2S} \cdot N^{2S-1}}{\alpha \cdot (W')^{0.5}}$$

This completes the proof of Theorem 4.

4.5.2 Example 5: Unfairness with Finite Windows

Consider a system that satisfies the assumptions of Chapter 2 and has the layout shown in Figure 11. The network contains links l_1 and l_2. (For each of these links, there is another link with opposite direction that is not shown in Figure 11 and is used only to return flow control permits.) Session x uses l_1 followed by l_2. Session y uses only l_1. Sessions z_1 and z_2 use only l_2. Sessions x and y have heavy demand; i.e.,

$$C(x, 0, t-1, t) = C(y, 0, t-1, t) = 1 \tag{269}$$

for all times $t \geq 1$. (Note that these demand processes are Bernoulli, with rate one.) Sessions z_1 and z_2 have independent Bernoulli demand processes with rate ¼; i.e.,

$$PROB\{ C(z_1, 0, t-1, t) = 1 \} = PROB\{ C(z_2, 0, t-1, t) = 1 \} = ¼ \tag{270}$$

for all times $t \geq 1$. The max-min fair rates for sessions x, y, z_1, and z_2 are ½, ½, ¼, and ¼, respectively. The window size for each buffer of each session is at least two. Round robin link scheduling is used.

Divide the time interval $(0, \infty)$ into subintervals of length $12 \cdot W(x, 2)$ slots, viz., $(0, 12 \cdot W(x, 2)]$, $(12 \cdot W(x, 2), 24 \cdot W(x, 2)]$, ... Since session y will accept every chance offered to it by the round robin scheduler at link l_1 (except possibly during slot 1), session x can transmit at most $6 \cdot W(x, 2)$ packets over l_1 during any of these subintervals. In other words, the average

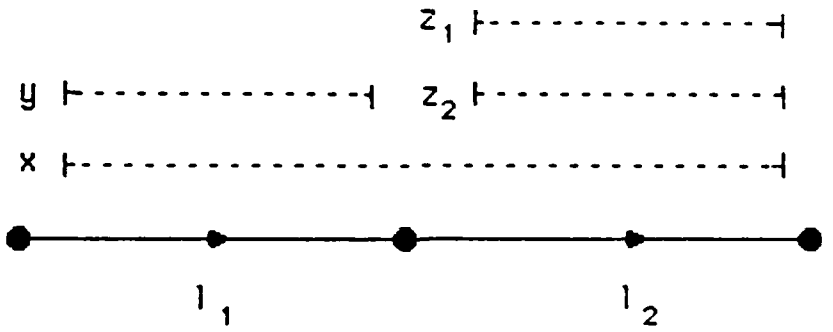


FIGURE 11

throughput of x over l_1 during any of these subintervals is at most ½ .

Things can be worse, however. Consider a particular subinterval $(12\cdot(k-1)\cdot W(x, 2)\ ,\ 12\cdot k\cdot W(x, 2)]$. Suppose that

$$C(z_1\ ,\ 0,\ t-1,\ t)\ =\ C(z_2\ ,\ 0,\ t-1,\ t)\ =\ 1 \tag{271}$$

for all slots t in this subinterval. (The probability of this event is $(¼\cdot¼)^{12\cdot W(x,2)} = (½)^{48\cdot W(x,2)}$.) Since sessions z_1 and z_2 will accept every chance offered to them by the round robin scheduler at link l_2 during the subinterval (except possibly during the first slot of the subinterval), session x can transmit no more than $4\cdot W(x, 2)$ packets over l_2 during the subinterval. Window flow control will permit session x to transmit at most $W(x, 2)$ more packets over link l_1 than x transmits over l_2 during the subinterval; this limits x to a total of $5\cdot W(x, 2)$ packets over l_1 . Therefore, the average throughput of x over l_1 during the subinterval is at most $5/12$.

In summary, during a fraction $(½)^{48\cdot W(x,2)}$ of the subintervals, the average throughput of session x over link l_1 is at most $5/12$, and during the other subintervals, the average throughput of x over l_1 is at most ½ . Therefore,

$$R_A(x) = \lim_{t \to \infty} \frac{P(x, 2, 0, t)}{t}$$

$$= \lim_{t \to \infty} \frac{P(x, 1, 0, t)}{t}$$

$$\leq \frac{5}{12} \cdot \left(\frac{1}{2}\right)^{48 \cdot W(x,2)} + \frac{1}{2} \cdot \left[1 - \left(\frac{1}{2}\right)^{48 \cdot W(x,2)}\right]$$

$$= \frac{1}{2} - \frac{1}{12} \cdot \left(\frac{1}{2}\right)^{48 \cdot W(x,2)}$$

$$= R_F(I(x)) - \frac{1}{12} \cdot \left(\frac{1}{2}\right)^{48 \cdot W(x,2)} \tag{272}$$

For no finite value of $W(x, 2)$ does the long-term average throughput $R_A(x)$ of session x equal its fair rate $R_F(I(x))$. This is due to the burstiness of the demands of sessions z_1 and z_2.

4.6 Unfairness with First-Come-First-Served Scheduling

The examples in this section demonstrate that if first-come-first-served link scheduling is used instead of round robin scheduling, then max-min throughput fairness is not guaranteed even if the windows are large and of comparable magnitude. In other words, Corollary 1 and Theorem 4 do not hold. These examples show that with first-come-first-served scheduling, the long-term average throughputs are strongly affected by the relative window sizes of competing sessions and by the initial conditions -- even if the windows are large. In Example 6, the capacity of a link shared by two sessions is divided between the sessions in proportion to their window sizes. Unfair average throughputs result if the sessions have unequal window sizes, no matter how large the windows are. Example 7 shows a complex system with $4N+4$ links and N^2+1 sessions, where N can be any even integer greater than ten. In this example, the sessions have equal window sizes. Some initial conditions result in fair average throughputs. For other initial conditions, however, the long-term average throughput of one session is unfair by a factor of more than $N/10$, no matter how large the window size. It seems that the problem of selecting window sizes to achieve throughput fairness, for general networks and general initial conditions, is difficult and perhaps impossible if first-come-first-served link scheduling is used.

4.6.1 Example 6: Unfairness with Unequal Windows

Consider a system that satisfies the assumptions of Chapter 2. The network consists of two nodes joined by links l_1 and l_2 of opposite direction. Link l_1 is used only by sessions x and y. Both sessions have heavy demand; i.e.,

$$C(x, 0, t-1, t) = C(y, 0, t-1, t) = 1 \tag{273}$$

for all times $t \geq 1$. Obviously, the max-min fair rate for each session is ½. Suppose that the window sizes $W(x, 1)$ and $W(x, 2)$ for session x equal w_x, the window sizes $W(y, 1)$ and $W(y, 2)$ for session y equal w_y, and $w_x \neq w_y$. The initial buffer levels are as follows:

$$B(x, 0, 0) = \infty \qquad B(x, 1, 0) = w_x - 1 \qquad B(x, 2, 0) = 1$$

$$B(y, 0, 0) = \infty \qquad B(y, 1, 0) = w_y \qquad B(y, 2, 0) = 0$$

First-come-first-served link scheduling is used. The tie-breaking list for link l_1 is arbitrary. The transmitter queue for l_1 initially contains w_x-1 reservations for x and w_y reservations for y, in arbitrary order.

The evolution of this system is very simple. During slot 1, session x transmits a packet over hop 2 (i.e., the packet is retrieved by the session's sink), the corresponding permit for buffer 2 is returned upstream to hop 1, and a new reservation for x is added to the tail of the transmitter queue at link l_1. Similarly, during the slot immediately following the transmission of a packet

over l_1, that packet is transmitted over hop 2, its permit for buffer 2 is returned upstream to hop 1, and a new reservation for that packet's session is added to l_1's transmitter queue. Hence link l_1 operates periodically, with period w_x+w_y. In each period, sessions x and y transmit w_x and w_y packets, respectively, over l_1. Therefore, the long-term average session throughputs are $w_x/(w_x+w_y)$ and $w_y/(w_x+w_y)$, respectively. Since $w_x \neq w_y$, these average throughputs are unfair. †

† It may seem that the conclusion of this example depends on the somewhat arbitrary way that session sources and sinks were modeled in Chapter 2. However, it is easy to embed this example in one with longer session paths, so that the interesting features occur at intermediate hops.

4.6.2 Example 7: Unfairness with Equal Windows

Let N be an even integer greater than ten. Consider a system that satisfies the assumptions of Chapter 2 and has the layout shown in Figure 12. For $i = 1, 2$ and $j = 1, 2, \ldots, \frac{1}{2}N$, the network contains links $l_{i,0}$, $l_{i,j,1}$, and $l_{i,j,2}$. (For each of these links, there is another link with opposite direction that is not shown in Figure 12 and is used only to return flow control permits.) For $i = 1, 2$ and $j = 1, 2, \ldots, \frac{1}{2}N$, there is a session $y_{i,j,1}$ that uses links $l_{i,j,1}$, $l_{i,j,2}$, and $l_{i,0}$, and there are $N-1$ sessions $y_{i,j,2}, y_{i,j,3}, \ldots, y_{i,j,N}$ that use only $l_{i,j,1}$. There is also a session x that uses $l_{1,0}$ followed by $l_{2,0}$. Every session in the system has heavy demand; i.e.,

$$C(x, 0, t-1, t) = C(y_{i,j,k}, 0, t-1, t) = 1 \tag{274}$$

for $i = 1, 2$, for $j = 1, 2, \ldots, \frac{1}{2}N$, for $k = 1, 2, \ldots, N$, and for all times $t \geq 1$. The max-min fair rate for session x is $1/2$, while the other sessions deserve rates of $1/N$ each. The windows for all buffers $h \geq 1$ of all sessions have the same size $w \geq 2$. Table 5 shows the buffer levels at time 0. First-come-first-served link scheduling is used. The tie-breaking lists are arbitrary. At time 0, the transmitter queue for link $l_{1,0}$ contains exactly one reservation each for sessions $y_{1,j,1}$, $j = 1, 2, \ldots, \frac{1}{2}N$, and possibly some reservations for session x. The transmitter queue for $l_{2,0}$ may initially contain any number of reservations for sessions x and $y_{2,j,1}$, $j = 1, 2, \ldots, \frac{1}{2}N$, as long as all reservations for sessions $y_{2,j,1}$ are in the first $2w$ queue positions. Let us

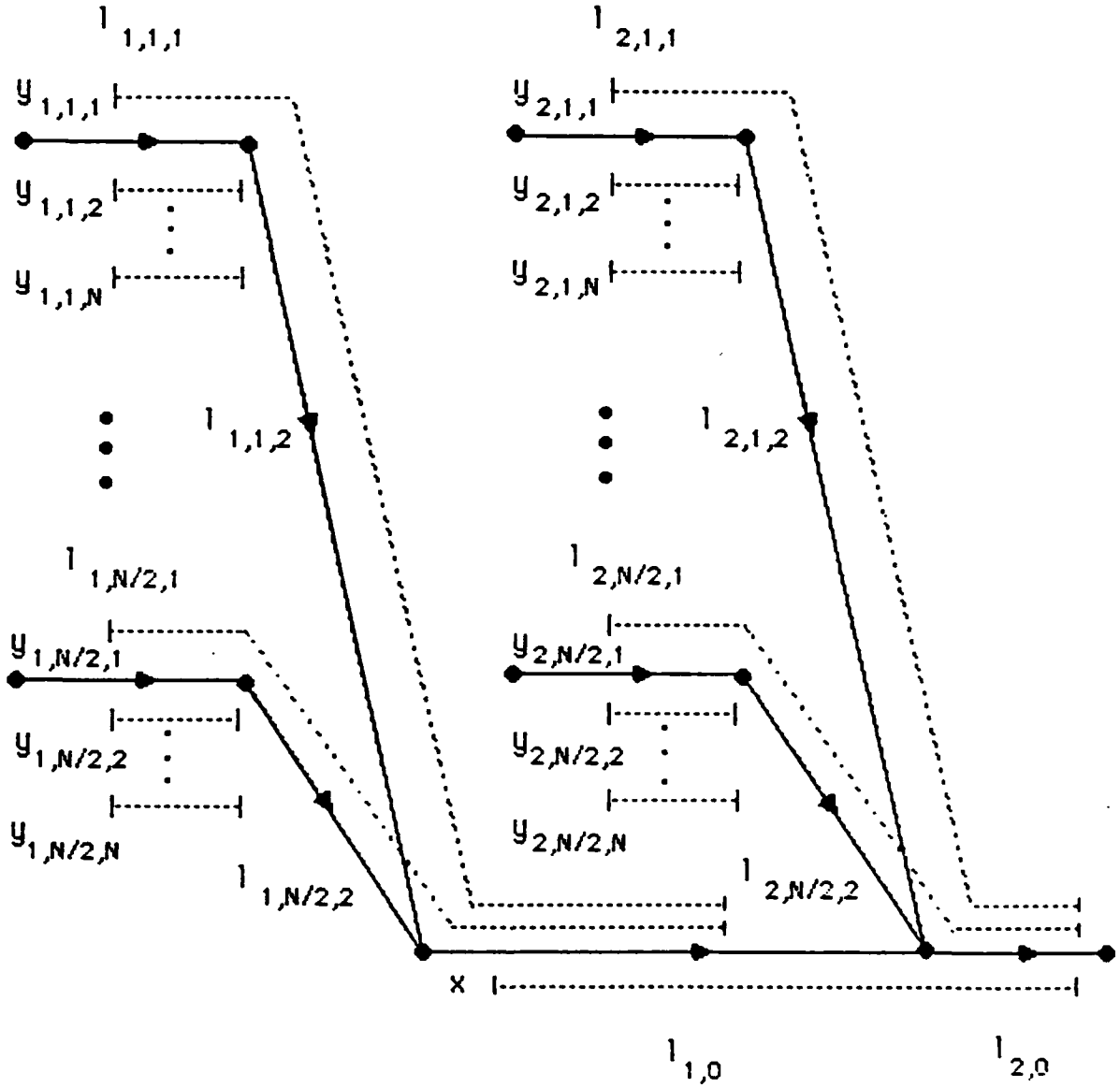

l 1,1,1
y 1,1,1
y 1,1,2
y 1,1,N
l 1,1,2
l 1,N/2,1
y 1,N/2,1
y 1,N/2,2
y 1,N/2,N
l 1,N/2,2
l 2,1,1
y 2,1,1
y 2,1,2
y 2,1,N
l 2,1,2
l 2,N/2,1
y 2,N/2,1
y 2,N/2,2
y 2,N/2,N
l 2,N/2,2
X
l 1,0
l 2,0

FIGURE 12

Session		Buffer Number				
		0	1	2	3	4
x		∞	w or $w-1$	between 0 and w	0 or 1	--
$y_{1,j,1}$	$1 \leq j \leq \frac{1}{2}N$	∞	$w-1$	1	1	0
$y_{1,j,k}$	$1 \leq j \leq \frac{1}{2}N$ $2 \leq k \leq N$	∞	w	0	--	--
$y_{2,j,1}$	$1 \leq j \leq \frac{1}{2}N$	∞	w	0	note below	0 or 1
$y_{2,j,k}$	$1 \leq j \leq \frac{1}{2}N$ $2 \leq k \leq N$ $k \neq \frac{1}{2}N + 1$	∞	w	0	--	--
$y_{2,j,\frac{1}{2}N+1}$	$1 \leq j \leq \frac{1}{2}N$	∞	$w-1$	1	--	--

Note: $\sum_{j=1}^{\frac{1}{2}N} B(y_{2,j,1}, 3, 0) \leq 2w$

TABLE 5. Initial Buffer Levels

now describe the initial transmitter queues at links $l_{1,j,1}$, $l_{1,j,2}$, $l_{2,j,1}$, and $l_{2,j,2}$, for $j = 1, 2, \ldots, \frac{1}{2}N$. The transmitter queue for $l_{1,j,1}$ contains $w-2$ reservations for $y_{1,j,1}$, followed by w reservations for $y_{1,j,2}$, then w reservations for $y_{1,j,3}$, ..., then w reservations for $y_{1,j,N}$, and finally one reservation for $y_{1,j,1}$. The transmitter queue for $l_{1,j,2}$ contains one reservation for $y_{1,j,1}$. The transmitter queue for $l_{2,j,1}$ contains $w-2$ reservations for $y_{2,j,\frac{1}{2}N+1}$, followed by w reservations for $y_{2,j,\frac{1}{2}N+2}$, ..., then w reservations for $y_{2,j,N}$, then w reservations for $y_{2,j,1}$, ..., then w reservations for $y_{2,j,\frac{1}{2}N}$, and finally one reservation for $y_{2,j,\frac{1}{2}N+1}$. The transmitter queue for $l_{2,j,2}$ is initially empty. †

This system will be analyzed over successive time intervals of length $N \cdot w$ slots. It will be shown that the throughput of session x during $[1, N \cdot w]$ is less than $5w$ packets. It will also be shown that the buffer levels and transmitter queues at time $N \cdot w$ satisfy the same assumptions that were made for their initial values, so that x's throughput bound during $[1, N \cdot w]$ also applies during $[N \cdot w+1, 2N \cdot w]$, $[2N \cdot w+1, 3N \cdot w]$, ... Table 6 shows which session

† In practice, this "initial" system state could arise if sessions $y_{i,j,k}$ started at various times *before* time 0, when there were already many *other* sessions using links $l_{1,j,1}$ and $l_{2,j,1}$, and if these extra sessions terminated before time 0.

uses each slot at each link during $[1, N{\cdot}w]$, for the case where $N = 12$ and $w = 3$.

Let us examine the operation of links $l_{1,0}$ and $l_{2,0}$ during the first half of the study interval $[1, N{\cdot}w]$. At time 0, the transmitter queue for $l_{1,0}$ contains at most w packets for session x and exactly one packet for each session $y_{1,j,1}$, $j = 1, 2, \ldots, \frac{1}{2}N$. During $[1, w-1]$, fewer than w packets for x and exactly $w-1$ packets for each session $y_{1,j,1}$ are added to this queue. In other words, the total number of packets to enter this queue by time $w-1$ is at least $\frac{1}{2}N{\cdot}w$ but fewer than $(\frac{1}{2}N+2){\cdot}w$. These packets will be transmitted over $l_{1,0}$ before any later arrivals for session x are served. Consequently, the packets for sessions $y_{1,j,1}$ are guaranteed to be transmitted over $l_{1,0}$ by time $(\frac{1}{2}N+2){\cdot}w$, well before the end $N{\cdot}w$ of the study interval. Furthermore, the throughput for x over $l_{1,0}$ during $[1, \frac{1}{2}N{\cdot}w]$ is limited to the (fewer than $2w$) packets joining $l_{1,0}$'s transmitter queue by time $w-1$. Therefore, the buffer capacity constraint will restrict x's transmission over $l_{2,0}$ during $[1, \frac{1}{2}N{\cdot}w]$ to fewer than $2w + w = 3w$ packets.

A similar analysis is possible for the second half of the study interval $[1, N{\cdot}w]$. It will be shown that intense competition from other sessions impedes session x's flow on its second link, while the window mechanism impedes the flow on its first link. During $[\frac{1}{2}N{\cdot}w-1, (\frac{1}{2}N+1){\cdot}w-2]$, link $l_{2,j,1}$ transmits w packets for session $y_{2,j,1}$, $j = 1, 2, \ldots, \frac{1}{2}N$. The initial conditions guarantee that by time $2w$ link $l_{2,0}$ finishes transmitting any

Slot	Link						
	$l_{1,j,1}$ $1 \le j \le 6$	$l_{1,j,2}$ $1 \le j \le 6$	$l_{1,0}$		$l_{2,0}$		$l_{2,j,2}$ $1 \le j \le 6$	$l_{2,j,1}$ $1 \le j \le 6$
1	$y_{1,j,1}$	$y_{1,j,1}$						$y_{2,j,7}$
2						?		
3	$y_{1,j,2}$					pkts		$y_{2,j,8}$
4						for		
5						$y_{2,j,1}$		
6	$y_{1,j,3}$							$y_{2,j,9}$
7								
8								
9	$y_{1,j,4}$						idle	$y_{2,j,10}$
10								
11			?	$<2w$	$<3w$	0		
12	$y_{1,j,5}$		pkts	pkts	pkts	pkts		$y_{2,j,11}$
13			for	for	for	for		
14			$y_{1,j,1}$	x	x	$y_{2,j,1}$		
15	$y_{1,j,6}$							$y_{2,j,12}$
16								
17								
18	$y_{1,j,7}$							$y_{2,j,1}$
19		idle					$y_{2,j,1}$	
20			?					
21	$y_{1,j,8}$		pkts					$y_{2,j,2}$
22			for					
23			$y_{1,j,1}$					
24	$y_{1,j,9}$							$y_{2,j,3}$
25								
26								
27	$y_{1,j,10}$							$y_{2,j,4}$
28							idle	
29			0	$<3w$	$<2w$	?		
30	$y_{1,j,11}$		pkts	pkts	pkts	pkts		$y_{2,j,5}$
31			for	for	for	for		
32			$y_{1,j,1}$	x	x	$y_{2,j,1}$		
33	$y_{1,j,12}$							$y_{2,j,6}$
34								
35	$y_{1,j,1}$							$y_{2,j,7}$
36		$y_{1,j,1}$						

TABLE 6. Link Users over One Period ($N = 12$, $w = 3$)

packets for session $y_{2,j,1}$ that were waiting initially. Therefore, a full supply of w permits for buffer 3 is available at hop 2 when the w new packets arrive over $l_{2,j,1}$, and these packets immediately join $l_{2,j,2}$'s transmitter queue. During $[\frac{1}{2}N\cdot w\ ,\ (\frac{1}{2}N+1)\cdot w-1]$, these packets are transmitted over $l_{2,j,2}$ and join the transmitter queue at $l_{2,0}$. At time $\frac{1}{2}N\cdot w$, there at most w packets for session x in $l_{2,0}$'s transmitter queue. During $[\frac{1}{2}N\cdot w+1\ ,\ (\frac{1}{2}N+1)\cdot w-1]$, fewer than w packets for x are added to the queue. These packets for sessions $y_{2,j,1}$ and x -- totalling at least $\frac{1}{2}N\cdot w$ packets but fewer than $(\frac{1}{2}N+2)\cdot w$ -- will be transmitted over $l_{2,0}$ starting at slot $\frac{1}{2}N\cdot w+1$, and they will all be transmitted before any later arrivals for session x are served. Consequently, the packets for sessions $y_{2,j,1}$ are guaranteed to be transmitted over $l_{2,0}$ by time $(N+2)\cdot w$. (In other words, any reservations for sessions $y_{2,j,1}$ in $l_{2,0}$'s transmitter queue at the end $N\cdot w$ of the study interval must be in the first $2w$ queue positions.) Furthermore, the throughput for x over $l_{2,0}$ during $[\frac{1}{2}N\cdot w+1\ ,\ N\cdot w]$ is limited to the (fewer than $2w$) packets that were present in $l_{2,0}$'s transmitter queue at time $\frac{1}{2}N\cdot w$ or joined it during $[\frac{1}{2}N\cdot w+1\ ,\ (\frac{1}{2}N+1)\cdot w-1]$. Therefore, the buffer capacity constraint restricts x's transmission over $l_{1,0}$ during $[\frac{1}{2}N\cdot w+1\ ,\ N\cdot w]$ to fewer than $3w$ packets.

This completes the analysis of links $l_{1,0}$ and $l_{2,0}$. The operation of links $l_{i,j,1}$ and $l_{i,j,2}$, $i = 1, 2$, $j = 1, 2, \ldots, \frac{1}{2}N$, during $[1, N\cdot w]$ is simpler. Link $l_{i,j,1}$ transmits a block of w packets for each session $y_{i,j,k}$ in turn.

During the slot immediately following a packet's transmission over $l_{i,j,1}$, that packet is transmitted over hop 2, its permit for buffer 2 is returned upstream to hop 1, and a new reservation is added to $l_{i,j,1}$'s transmitter queue. (For session $y_{i,j,1}$, hop 2 is link $l_{i,j,2}$. For each session $y_{i,j,k}$, $k \neq 1$, hop 2 is the session's sink.) Recall that $y_{i,j,1}$'s packets are transmitted over $l_{i,0}$ in plenty of time to get their permits for buffer 3 back to hop 2 before the next batch of packets for $y_{i,j,1}$ arrive over $l_{i,j,1}$. Hence the transmitter queues for links $l_{i,j,1}$ and $l_{i,j,2}$ are periodic, with period $N \cdot w$.

In summary, over the entire study interval $[1, N \cdot w]$ each session $y_{i,j,k}$, $i = 1, 2$, $j = 1, 2, \ldots, \frac{1}{2}N$, $k = 1, 2, \ldots, N$, transmits exactly w packets over link $l_{i,j,1}$, while session x transmits fewer than $2w + 3w = 5w$ packets over each of its links. Since the system satisfies the initial condition assumptions again at time $N \cdot w$, these throughput claims also hold for time intervals $[N \cdot w + 1, 2N \cdot w]$, $[2N \cdot w + 1, 3N \cdot w]$, ... Therefore, the long-term average throughputs of sessions $y_{i,j,k}$ are max-min fair, but the long-term average throughput of x is less than $5/N$, which is significantly lower than its fair rate of $1/2$. † In other words, x's average throughput is unfair by a factor of more than $N/10$, regardless of the window size w. Moreover, the capacity

† The long-term average session throughputs must exist because this system has a finite number of states and is deterministic. Eventually the system will enter some state it has already visited, after which the system will be periodic.

lost by x (viz., more than $1/2 - 5/N$ at each of its links) is not used by the other sessions -- it is wasted. †

The unfairness in this example depends critically on the unfortunate initial conditions. Other initial conditions for this same network can result in fair average throughputs. For example, suppose that the transmitter queue for link $l_{1,0}$ at time 0 contains w reservations for session x alternating with $w-1$ reservations for session $y_{1,½N,1}$. The transmitter queue for $l_{2,0}$ initially contains one reservation for $y_{2,½N,1}$. The initial transmitter queues for the other links are arranged so that packets for session $y_{i,j,1}$, $i = 1, 2$, $j = 1, 2, \ldots, ½N$, are transmitted over link $l_{i,j,2}$ during slots $(i+2w\cdot j-2w)$, $(i+2w\cdot j-2w+2)$, $(i+2w\cdot j-2w+4), \ldots, (i+2w\cdot j-2)$. In other words, packets competing with session x arrive in smooth streams, during the odd-numbered time slots at link $l_{1,0}$ and during the even-numbered slots at $l_{2,0}$. Therefore, during the interval $[1, N\cdot w]$, session x transmits one packet across $l_{1,0}$ during each odd-numbered slot and one packet across $l_{2,0}$ during each even-numbered slot. By correctly setting the initial conditions, the entire system can be made periodic with period $N\cdot w$, so that these smooth flows continue forever and the long-term average throughputs are fair.

---

† It may seem that the conclusion of this example depends on the somewhat arbitrary way that session sources and sinks were modeled in Chapter 2. However, it is easy to embed this example in one with longer session paths, so that the interesting features occur at intermediate hops.

5. SESSION THROUGHPUTS IN SYSTEMS WITH SMALL WINDOWS

This chapter studies the throughput of a particular session x in a system with bounded-delay link scheduling. The window sizes for session x are assumed to be at least two. Except for $W(x, 0)$ and possibly $W(x, 1)$, the windows for session x are assumed to be finite. The window sizes of the other sessions in the network are arbitrary. Session x is assumed to have a well-defined, real demand rate $\lambda(x)$ in the range $0 < \lambda(x) \leq 1$. The detailed demand assumptions for x vary from section to section. The demands of the other sessions are arbitrary, except for the possible restriction that these demands be independent of the demand of session x. These other sessions need not have well-defined demand rates. Clearly, the assumptions of this chapter are much less restrictive than those of Chapter 4. The results of this chapter, therefore, are of greater practical value.

Theorem 5 assumes that $W(x, 1)$ is at least two but finite. The demand of session x is modeled as a Bernoulli process that is independent of the other sessions' demands. The theorem concludes that the throughput $P(x, H(x), 0, t)$ of x is bounded below by a function $\Phi(t)$ whose long-term average rate equals (with probability one) $\frac{\lambda(x)}{[1-\lambda(x)]^{A(x)} + A(x)\cdot\lambda(x)}$. As one would expect, this guaranteed rate tends to zero as the demand rate $\lambda(x)$ tends to zero or as the schedule delay bound $A(x)$ tends to infinity. As $\lambda(x)$ tends to one, the guaranteed rate tends to $1/A(x)$, and as $A(x)$ tends to

one, the guaranteed rate tends to $\lambda(x)$; these limits are also intuitive.

Theorem 6 shows that the guaranteed rate can be increased by allowing session x to buffer more of its demand: this theorem assumes $W(x,1)$ to be infinite. The demand assumptions are a little weaker than before: the times between chances for session x at hop 0 are only assumed to be independent and identically distributed, and the demand of session x may be dependent on the demands of the other sessions. Theorem 6 concludes that the throughput $P(x, H(x), 0, t)$ of x is bounded below by a function $\Phi(t)$ whose long-term average rate equals (with probability one) $\min[1/A(x), \lambda(x)]$. This is obviously the largest rate guarantee possible if nothing else is known about the scheduling discipline; a better guarantee cannot be achieved in general by assuming window sizes larger than two.

It was explained in Section 2.4.3 that round robin scheduling and first-come-first-served scheduling are bounded delay disciplines, with schedule delay bounds $A'(l)$ of $N'(l)$ and $N'(l)\cdot W'' - W'' + 1$, respectively. Therefore, the rate guarantees of Theorem 5 for round robin systems and first-come-first-served systems are $\dfrac{\lambda(x)}{[1-\lambda(x)]^{N(x)} + N(x)\cdot\lambda(x)}$ and $\dfrac{\lambda(x)}{[1-\lambda(x)]^{N(x)\cdot W'' - W'' + 1} + [N(x)\cdot W'' - W'' + 1]\cdot\lambda(x)}$, respectively. The rate guarantees of Theorem 6 are $\min[1/N(x), \lambda(x)]$ and $\min[1/(N(x)\cdot W'' - W'' + 1), \lambda(x)]$, respectively. Example 1 of Section 3.2 shows a round robin system with a session x whose demand rate $\lambda(x) = 1$ and

whose long-term average throughput is $1/N(x)$. Example 2 of Section 3.3 shows a first-come-first-served system with a session x whose demand rate $\lambda(x) = 1$ and whose long-term average throughput is $1/[N(x) \cdot W'' - W'' + 1]$. The average throughputs in these examples match the bounds of Theorems 5 and 6. Note that the throughput guarantees of Theorems 5 and 6 for round robin systems are superior to those for first-come-first-served systems. (It is *not* being claimed that round robin scheduling *always* offers larger throughputs or fairer throughputs than first-come-first-served scheduling.)

Let us compare the throughput guarantees of Theorem 6 to the max-min fair rates defined in Section 4.1. Consider a system that satisfies the assumptions of Theorem 6 and has a demand process so regular that the long-term average throughputs $R_A(x)$, the demand rates $\lambda(x)$, and hence the max-min fair rates $R_F(I(x))$ exist for all sessions x. Obviously, for each session x, $R_A(x) \leq \lambda(x) \leq 1$ and $R_F(I(x)) \leq \lambda(x) \leq 1$. Recall from Section 4.1 that every session x has at least one bottleneck hop. By definitions (64) and (65), this means that either $R_F(I(x)) = \lambda(x)$ or x has a bottleneck link l such that

$$R_F(I(x)) = 1 - \sum_{\substack{y \text{ using } l \\ y \neq x}} R_F(I(y))$$

$$\geq 1 - \sum_{\substack{y \text{ using } l \\ y \neq x}} R_F(I(x))$$

$$= 1 - [N'(l) - 1] \cdot R_F(I(x))$$

$$\geq 1 - [N(x) - 1] \cdot R_F(I(x)) \tag{275}$$

and hence $R_F(I(x)) \geq 1/N(x)$. If the system uses round robin link scheduling, Theorem 6 guarantees that $R_A(x) \geq \min[1/N(x), \lambda(x)]$. Combining these various results shows that if $\lambda(x) < 1/N(x)$, then $R_F(I(x)) = \lambda(x) = R_A(x)$, and if $\lambda(x) \geq 1/N(x)$, then $1/N(x) \leq R_F(I(x)) \leq 1$ and $1/N(x) \leq R_A(x) \leq 1$ and hence $\frac{1}{N(x)} \leq \frac{R_F(I(x))}{R_A(x)} \leq N(x)$. In either case, the long-term average throughput of x is within a factor of $N(x)$ of its fair rate. The analogous guarantee for first-come-first-served scheduling involves a factor of $N(x) \cdot W'' - W'' + 1$. Example 3 of Section 4.4.2 shows that, with round robin scheduling, unfairness factors proportional to $N(x)$ are actually possible. For first-come-first-served scheduling, Example 7 of Section 4.6.2 shows an unfairness factor proportional

to $N(x)$ and Example 2 of Section 3.3 shows an unfairness factor roughly equal to W'' . †

† For ease of exposition, it was assumed in Examples 3 and 7 that $W(x,1)$ is finite. However, the long-term average session throughputs for these examples are the same whether $W(x, 1)$ is finite or infinite.

5.1 Theorem 5: Throughput Bound, given Finite Demand Buffer

Suppose a system has been specified that satisfies the assumptions of Chapter 2 and uses a bounded delay scheduling discipline. Let x be some session. Suppose that

$$(276) \qquad 2 \leq W(x, h) < \infty \qquad \text{for } 1 \leq h \leq H(x)+1$$

The window sizes of the other sessions in the network are arbitrary (i.e., these window sizes only need to satisfy the basic assumptions of Chapter 2). Suppose that chances for session x at hop 0 form a Bernoulli process with rate $\lambda(x)$, and $0 < \lambda(x) \leq 1$. Suppose that the demand of session x is independent of the demands of the other sessions in the network; except for this restriction, the demands of the other sessions are arbitrary. It follows that there exist random variables $\Phi(1), \Phi(2), \ldots$ such that, for all times $t \geq 1$,

$$(277) \qquad P(x, H(x), 0, t) \geq \Phi(t)$$

and such that, with probability one,

$$(278) \qquad \frac{\Phi(t)}{t} \xrightarrow[t \to \infty]{} \frac{\lambda(x)}{[1 - \lambda(x)]^{A(x)} + A(x) \cdot \lambda(x)}$$

In other words, the average throughput $P(x, H(x), 0, t)/t$ of session x can be made arbitrarily close to $\frac{\lambda(x)}{[1 - \lambda(x)]^{A(x)} + A(x) \cdot \lambda(x)}$ -- or greater -- by taking t sufficiently large. (This cannot be phrased in terms of the long-term average throughput of x; i.e., we *cannot* say:

$$R_A(x) = \lim_{t \to \infty} \frac{P(x, H(x), 0, t)}{t} \geq \frac{\lambda(x)}{[1 - \lambda(x)]^{A(x)} + A(x) \cdot \lambda(x)}$$

since the limit above may not exist.)

Proof of Theorem 5

First let us clarify the scheduling assumptions. It follows from (21) and (276) that, for all packets $p \geq 1$ of x and all hops h in the range $1 \leq h \leq H(x)$,

$$\begin{aligned}
&\Upsilon(x, h, p) \\
&\leq \max [\, \Upsilon(x, h-1, p), \Upsilon(x, h, p-1), \Upsilon(x, h+1, p - W(x, h+1)) \,] + A(x, h) \\
&\leq \max [\, \Upsilon(x, h-1, p), \Upsilon(x, h, p-1), \Upsilon(x, h+1, p - W(x, h+1)) \,] + A(x) \\
&\leq \max [\, \Upsilon(x, h-1, p), \Upsilon(x, h, p-1), \Upsilon(x, h+1, p-2) \,] + A(x) \qquad (279)
\end{aligned}$$

Next the demand assumptions will be clarified. Let p' be the first packet of session x to be transmitted over hop 0 after time 0. For all packets $p \geq 1$, define τ_p as follows.

$$(280) \qquad \tau_p = \Upsilon(x, 0, p) - \max [\, \Upsilon(x, 0, p-1), \Upsilon(x, 1, p - W(x, 1)) \,]$$

Note that $\tau_p = 0$ for $1 \leq p < p'$. For $p \geq p'$, τ_p is the delay from the time the network is prepared to accept packet p (i.e., when all packets older than p have been transmitted over hop 0 and there is room in buffer 1) until

packet p is actually transmitted over hop 0. Since the demand of session x is Bernoulli with rate $\lambda(x)$ and is independent of the other sessions' demands, it follows that $\tau_{p'}$, $\tau_{p'+1}$, ... are independent and identically distributed according to a geometric distribution with mean $1/\lambda(x)$:

$$(281) \qquad PROB\{\tau_p = k\} = \lambda(x)\cdot[1-\lambda(x)]^{k-1} \qquad \text{for } p \geq p',\ k \geq 1$$

For each hop h of x in the range $0 \leq h \leq H(x)$ and every integer p, define $\Theta(h, p)$ as follows.

$$(282) \qquad \Theta(h, p) = \begin{cases} \sum_{q=1}^{p} \max[\tau_q, A(x)] + h\cdot A(x) & \text{for } p \geq 1 \\ 0 & \text{for } p \leq 0 \end{cases}$$

Note that

$$(283) \qquad \Theta(h-1, p) + A(x) = \Theta(h, p) \qquad \text{for } 1 \leq h \leq H(x),\ p \geq 1$$

and

$$(284) \qquad \Theta(0, p-1) + \tau_p \leq \Theta(0, p) \qquad \text{for } p \geq 1$$

and

$$(285) \qquad \Theta(h-1, p) \geq \Theta(h, p-1) \qquad \text{for } 1 \leq h \leq H(x),\ p \geq 0$$

It follows from (281) that

$$(286) \qquad EXPECTATION\{\max[\tau_p, A(x)]\} = \frac{1 - [1-\lambda(x)]^{A(x)}}{\lambda(x)} + A(x)[1-\lambda(x)]^{A(x)} \qquad \text{for } p \geq p'$$

It follows from (282), the strong law of large numbers [3], and (286) that, with

probability one, $\lim_{p \to \infty} \Theta(H(x), p)/p$ exists and

$$\lim_{p \to \infty} \frac{\Theta(H(x), p)}{p} = \lim_{p \to \infty} \frac{\sum_{q=1}^{p} \max[\tau_q , A(x)]}{p}$$

$$= \lim_{p \to \infty} \frac{\sum_{q=p'}^{p} \max[\tau_q , A(x)]}{p}$$

$$= \lim_{(p-p'+1) \to \infty} \frac{\sum_{q=p'}^{p} \max[\tau_q , A(x)]}{p - p' + 1}$$

$$(287) \qquad = \frac{[1 - \lambda(x)]^{A(x)} + A(x) \cdot \lambda(x)}{\lambda(x)}$$

Let us prove the following claim:

$$(288) \qquad \Upsilon(x, h, p) \leq \Theta(h, p) \quad \text{for } 0 \leq h \leq H(x), \ p \geq -1$$

The proof is by induction on p. The base cases $p = -1$ and $p = 0$ are trivial:

$$(289) \qquad \Upsilon(x, h, -1) = 0 = \Theta(h, -1) \quad \text{for } 0 \leq h \leq H(x)$$

$$(290) \qquad \Upsilon(x, h, 0) = 0 = \Theta(h, 0) \quad \text{for } 0 \leq h \leq H(x)$$

For the induction step, we fix a value $p \geq 1$. The induction hypothesis assert that

(291) $$\Upsilon(x, h, \hat{p}-2) \leq \Theta(h, \hat{p}-2) \quad \text{for } 0 \leq h \leq H(x)$$

and

(292) $$\Upsilon(x, h, \hat{p}-1) \leq \Theta(h, \hat{p}-1) \quad \text{for } 0 \leq h \leq H(x)$$

It must be shown that

(293) $$\Upsilon(x, h, \hat{p}) \leq \Theta(h, \hat{p}) \quad \text{for } 0 \leq h \leq H(x)$$

The proof of the induction step will itself be an induction, this time over h. For the base case (i.e., $h = 0$), first apply (280) and (276):

$$\Upsilon(x, 0, \hat{p}) = \max [\, \Upsilon(x, 0, \hat{p}-1), \Upsilon(x, 1, \hat{p} - W(x, 1)) \,] + \tau_{\hat{p}}$$

(294) $$\leq \max [\, \Upsilon(x, 0, \hat{p}-1), \Upsilon(x, 1, \hat{p}-2) \,] + \tau_{\hat{p}}$$

Now apply (292), (291), (285), and (284) to (294) to reach the desired conclusion:

$$\Upsilon(x, 0, \hat{p}) \leq \max [\; \Theta(0, \hat{p}-1), \quad \Theta(1, \hat{p}-2) \;] + \tau_{\hat{p}}$$

$$= \Theta(0, \hat{p}-1) + \tau_{\hat{p}}$$

(295) $$\leq \Theta(0, \hat{p})$$

For the induction step, consider a hop $\hat{h}$ of x in the range $1 \leq \hat{h} < H(x)$. (The case $\hat{h} = H(x)$ will be treated separately.) The induction hypothesis asserts that

(296) $$\Upsilon(x, \hat{h}-1, \hat{p}) \leq \Theta(\hat{h}-1, \hat{p})$$

It must be shown that

(297) $$\Upsilon(x, \hat{h}, \hat{p}) \leq \Theta(\hat{h}, \hat{p})$$

From (279), induction hypothesis (296) (for the induction on h), and induction hypotheses (292) and (291) (for the induction on p), it follows that

(298) $$\Upsilon(x, \hat{h}, \hat{p}) \leq \max [\, \Upsilon(x, \hat{h}-1, \hat{p}), \Upsilon(x, \hat{h}, \hat{p}-1), \Upsilon(x, \hat{h}+1, \hat{p}-2) \,] + A(x)$$

(299) $$\leq \max [\, \Theta(\hat{h}-1, \hat{p}), \Theta(\hat{h}, \hat{p}-1), \Theta(\hat{h}+1, \hat{p}-2) \,] + A(x)$$

Applying (285) and (283) to (299) gives the desired result (297):

$$\Upsilon(x, \hat{h}, \hat{p}) \leq \Theta(\hat{h}-1, \hat{p}) + A(x)$$

$$= \Theta(\hat{h}, \hat{p})$$

The proof for the remaining case, viz., $\hat{h} = H(x)$, is similar, but inequality (12) must be used to handle the term $\Upsilon(x, \hat{h}+1, \hat{p}-2)$ in (298) above. The proof of this case will not be presented. This completes the proof of (293) by induction on h, thereby completing the proof of (288) by induction on p.

For all times $t \geq 1$, define $\Phi(t)$ as follows:

(300) $$\Phi(t) = \max \{\, p: \Theta(H(x), p) \leq t \,\}$$

Note that (277) follows from (300), (288) and the fact that

(301) $$P(x, H(x), 0, t) = \max \{\, p: \Upsilon(x, H(x), p) \leq t \,\}$$

Also note that, for all times $t \geq 1$,

$$\Theta(H(x), \Phi(t)+1) \geq t \geq \Theta(H(x), \Phi(t)) \tag{302}$$

and hence

$$\frac{[\Phi(t)+1]-1}{\Theta(H(x), \Phi(t)+1)} \leq \frac{\Phi(t)}{t} \leq \frac{\Phi(t)}{\Theta(H(x), \Phi(t))} \tag{303}$$

Since $\Phi(t) \xrightarrow[t\to\infty]{} \infty$ and $\Theta(H(x), p) \xrightarrow[p\to\infty]{} \infty$, it follows from (303) and (287) that, with probability one, $\lim_{t\to\infty} \Phi(t)/t$ exists and

$$\lim_{t\to\infty} \frac{\Phi(t)}{t} = \frac{1}{\lim_{p\to\infty} \frac{\Theta(H(x), p)}{p}}$$

$$= \frac{\lambda(x)}{[1-\lambda(x)]^{A(x)} + A(x)\cdot\lambda(x)}$$

This is the desired result (278), completing the proof of Theorem 5.

5.2 Theorem 6: Throughput Bound, given Infinite Demand Buffer

Suppose a system has been specified that satisfies the assumptions of Chapter 2 and uses a bounded delay scheduling discipline. Let x be some session. Suppose that buffer 1 of x has infinite capacity:

(304) $$W(x, 1) = \infty$$

but that

(305) $$2 \leq W(x, h) < \infty \quad \text{for } 2 \leq h \leq H(x)+1$$

The window sizes of the other sessions in the network are arbitrary (i.e., these window sizes only need to satisfy the basic assumptions of Chapter 2). Suppose that the times between chances for session x at hop 0 are independent and identically distributed, with mean $1/\lambda(x)$, and $0 < \lambda(x) \leq 1$. (Because of (304), each such chance will result in a packet transmission.) The demands of the other sessions in the network are arbitrary. It follows that there exist random variables $\Phi(1), \Phi(2), \ldots$ such that, for all times $t \geq 1$,

(306) $$P(x, H(x), 0, t) \geq \Phi(t)$$

and such that, with probability one,

(307) $$\frac{\Phi(t)}{t} \xrightarrow[t \to \infty]{} \min\left[\frac{1}{\lambda(x)}, \lambda(x)\right]$$

In other words, the average throughput $P(x, H(x), 0, t)/t$ of session x can be made arbitrarily close to $\min[1/\lambda(x), \lambda(x)]$ -- or greater -- by taking t

sufficiently large. (This cannot be phrased in terms of the long-term average throughput of x; i.e., we *cannot* say:

$$R_A(x) = \lim_{t \to \infty} \frac{P(x, H(x), 0, t)}{t} \geq \min \left[\frac{1}{A(x)} , \lambda(x) \right]$$

since the limit above may not exist.)

Proof of Theorem 6

First let us clarify the scheduling assumptions. It follows from (21) and (305) that, for all packets $p \geq 1$ of x and all hops h in the range $1 \leq h \leq H(x)$,

$$\begin{aligned} &\Upsilon(x, h, p) \\ &\leq \max [\Upsilon(x, h-1, p), \Upsilon(x, h, p-1), \Upsilon(x, h+1, p-W(x, h+1))] + A(x, h \\ &\leq \max [\Upsilon(x, h-1, p), \Upsilon(x, h, p-1), \Upsilon(x, h+1, p-W(x, h+1))] + A(x) \\ &\leq \max [\Upsilon(x, h-1, p), \Upsilon(x, h, p-1), \Upsilon(x, h+1, p-2)] + A(x) \end{aligned} \tag{308}$$

Next the demand assumptions will be clarified. Let p' be the first packet of session x to be transmitted over hop 0 after time 0: i.e.,

$$\Upsilon(x, 0, p) = 0 \quad \text{for } 1 \leq p < p' \tag{309}$$

$$\Upsilon(x, 0, p) > 0 \quad \text{for } p \geq p' \tag{310}$$

For $p \geq p'+1$, let τ_p be the delay between the transmission times at hop 0

of packets $p-1$ and p for session x :

(311) $$\tau_p = \Upsilon(x, 0, p) - \Upsilon(x, 0, p-1) \quad \text{for } p \geq p'+1$$

It is given that $\Upsilon(x, 0, p')$, $\tau_{p'+1}$, $\tau_{p'+2}$, ... are independent and that $\tau_{p'+1}$, $\tau_{p'+2}$, ... are identically distributed, with mean $1/\lambda(x)$. (Note that the time $\Upsilon(x, 0, p')$ until the first transmission over hop 0 may have a different distribution and a different mean than the intertransmission times $\tau_{p'+1}$, $\tau_{p'+2}$, ...) Also define

(312) $$\tau_p = A(x) \quad \text{for } 1 \leq p \leq p'$$

It follows from (311) and (312) that, for all packets $p \geq 1$,

(313) $$\Upsilon(x, 0, p) \leq \Upsilon(x, 0, p') + \tau_1 + \tau_2 + \cdots + \tau_p$$

For each hop h of x in the range $0 \leq h \leq H(x)$ and every integer p, define $\Theta(h, p)$ as follows.

(314) $$\Theta(h, p) = \begin{cases} \Upsilon(x, 0, p') + (p + h)\cdot A(x) + \max\limits_{1 \leq J \leq p} \sum\limits_{j=1}^{J} [\tau_j - A(x)] \\ 0 \end{cases}$$

Note that

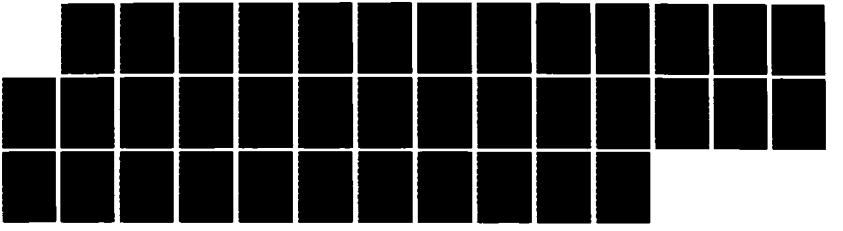
(315) $$\Theta(h-1, p) + A(x) = \Theta(h, p)$$

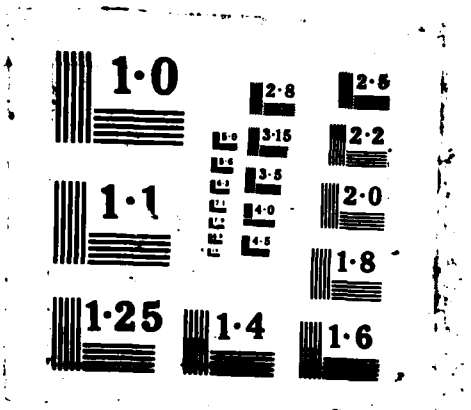
and

(316) $$\Theta(h-1, p) \geq$$

AD-A176 864 ROUND ROBIN SCHEDULING FOR FAIR FLOW CONTROL IN DATA COMMUNICATION NETWOR..(U) MASSACHUSETTS INST OF TECH CAMBRIDGE LAB FOR INFORMATION AND D.. E L HAHNE DEC 86 3/3

UNCLASSIFIED LIDS-TH-1631 N00014-84-K-0357 F/G 17/2 NL




1·0
1·1
1·25
1·4
1·6
1·8
2·0
2·2
2·5
2·8
3·15
3·5
4·0
4·5

It follows from definitions (314) and (312) that, for all $p \geq p'$,

$$\Theta(H(x), p) = \Upsilon(x, 0, p') + [p + H(x)]\cdot A(x) + \max_{1 \leq J \leq p} \sum_{j=p'+1}^{J} [\tau_j - A(x)]$$

$$= \Upsilon(x, 0, p') + [p + H(x)]\cdot A(x) + \max_{p' \leq J \leq p} \sum_{j=p'+1}^{J} [\tau_j - A(x)]$$

$$= \Upsilon(x, 0, p') + [p + H(x)]\cdot A(x) + \max_{0 \leq K \leq p-p'} \sum_{k=1}^{K} [\tau_{p'+k} - A(x)] \tag{317}$$

From (317) and Lemma 10 of Appendix A.3 it follows that, with probability one, $\lim_{p \to \infty} \Theta(H(x), p)/p$ exists and

$$\lim_{p\to\infty}\frac{\Theta(H(x),p)}{p} = A(x) + \lim_{p\to\infty}\frac{\max_{0\le K\le p-p'}\sum_{k=1}^{K}[\tau_{p'+k} - A(x)]}{p}$$

$$= A(x) + \lim_{(p-p')\to\infty}\frac{\max_{0\le K\le p-p'}\sum_{k=1}^{K}[\tau_{p'+k} - A(x)]}{p - p'}$$

$$= A(x) + \lim_{(p-p')\to\infty}\max_{0\le K\le p-p'}\frac{\sum_{k=1}^{K}[\tau_{p'+k} - A(x)]}{p - p'}$$

$$= A(x) + \max\left[0, \left(\frac{1}{\lambda(x)} - A(x)\right)\right]$$

$$= \max\left[A(x), \frac{1}{\lambda(x)}\right] \tag{318}$$

Let us prove the following claim:

$$\Upsilon(x,h,p) \le \Theta(h,p) \quad \text{for } 0 \le h \le H(x),\ p \ge -1 \tag{319}$$

The proof is by induction on p. The base cases $p = -1$ and $p = 0$ are trivial:

$$\Upsilon(x,h,-1) = 0 = \Theta(h,-1) \quad \text{for } 0 \le h \le H(x) \tag{320}$$

$$\Upsilon(x,h,0) = 0 = \Theta(h,0) \quad \text{for } 0 \le h \le H(x) \tag{321}$$

For the induction step, consider a packet $\hat{p} \ge 1$. The induction hypotheses assert that

(322) $$\Upsilon(x, h, \hat{p}-2) \leq \Theta(h, \hat{p}-2) \quad \text{for } 0 \leq h \leq H(x)$$

and

(323) $$\Upsilon(x, h, \hat{p}-1) \leq \Theta(h, \hat{p}-1) \quad \text{for } 0 \leq h \leq H(x)$$

It must be shown that

(324) $$\Upsilon(x, h, \hat{p}) \leq \Theta(h, \hat{p}) \quad \text{for } 0 \leq h \leq H(x)$$

The proof of the induction step will itself be an induction, this time over h. The base case (i.e., $h = 0$) follows from (313) and definition (314):

$$\Upsilon(x, 0, \hat{p}) \leq \Upsilon(x, 0, p') + \tau_1 + \tau_2 + \cdots + \tau_{\hat{p}}$$

$$= \Upsilon(x, 0, p') + \hat{p} \cdot A(x) + \sum_{j=1}^{\hat{p}} [\tau_j - A(x)]$$

$$\leq \Upsilon(x, 0, p') + \hat{p} \cdot A(x) + \max_{1 \leq J \leq \hat{p}} \sum_{j=1}^{J} [\tau_j - A(x)]$$

(325) $$= \Theta(0, \hat{p})$$

For the induction step, consider a hop $\hat{h}$ of x in the range $1 \leq \hat{h} < H(x)$. (The case $\hat{h} = H(x)$ will be treated separately.) The induction hypothesis asserts that

(326) $$\Upsilon(x, \hat{h}-1, \hat{p}) \leq \Theta(\hat{h}-1, \hat{p})$$

It must be shown that

(327) $$\Upsilon(x, \hat{h}, \hat{p}) \leq \Theta(\hat{h}, \hat{p})$$

From (308), induction hypothesis (326) (for the induction on h), and induction hypotheses (323) and (322) (for the induction on p), it follows that

(328) $$\Upsilon(x, \hat{h}, \hat{p}) \leq \max [\, \Upsilon(x, \hat{h}-1, \hat{p}), \Upsilon(x, \hat{h}, \hat{p}-1), \Upsilon(x, \hat{h}+1, \hat{p}-2) \,] + A(x)$$

(329) $$\leq \max [\quad \Theta(\hat{h}-1, \hat{p}), \quad \Theta(\hat{h}, \hat{p}-1), \quad \Theta(\hat{h}+1, \hat{p}-2) \quad] + A(x)$$

Applying (316) and (315) to (329) gives the desired result (327):

$$\Upsilon(x, \hat{h}, \hat{p}) \leq \Theta(\hat{h}-1, \hat{p}) + A(x)$$

$$= \Theta(\hat{h}, \hat{p})$$

The proof for the remaining case, viz., $\hat{h} = H(x)$, is similar, but inequality (12) must be used to handle the term $\Upsilon(x, \hat{h}+1, \hat{p}-2)$ in (328) above. The proof of this case will not be presented. This completes the proof of (324) by induction on h, thereby completing the proof of (319) by induction on p.

For all times $t \geq 1$, define $\Phi(t)$ as follows:

(330) $$\Phi(t) = \max \{\, p\colon\ \Theta(H(x), p) \leq t \,\}$$

Note that (306) follows from (330), (319) and the fact that

(331) $$P(x, H(x), 0, t) = \max \{\, p\colon\ \Upsilon(x, H(x), p) \leq t \,\}$$

Also note that, for all times $t \geq 1$,

(332) $$\Theta(H(x), \Phi(t)+1) \geq t \geq \Theta(H(x), \Phi(t))$$

and hence

$$(333) \qquad \frac{[\Phi(t)+1]-1}{\Theta(H(x),\Phi(t)+1)} \leq \frac{\Phi(t)}{t} \leq \frac{\Phi(t)}{\Theta(H(x),\Phi(t))}$$

Since $\Phi(t) \xrightarrow[t\to\infty]{} \infty$ and $\Theta(H(x),p) \xrightarrow[p\to\infty]{} \infty$, it follows from (333) and (318) that, with probability one, $\lim_{t\to\infty} \Phi(t)/t$ exists and

$$\lim_{t\to\infty} \frac{\Phi(t)}{t} = \frac{1}{\lim_{p\to\infty} \frac{\Theta(H(x),p)}{p}}$$

$$= \frac{1}{\max\left[A(x), \frac{1}{\lambda(x)}\right]}$$

$$= \min\left[\frac{1}{A(x)}, \lambda(x)\right]$$

This is the desired result (307), completing the proof of Theorem 6.

6. CONCLUSIONS

Round robin scheduling with windows is a simple technique for allocating link capacity among competing sessions in a packet network. If a sufficiently large window size is used throughout the network, the session throughput rates can be made arbitrarily close to the ideal max-min fair rates. (A session requiring a very large throughput rate could be visited more than once in each polling cycle, effectively treating it like several standard sessions.) This performance is suited to file transfers and other applications where throughput rate is of greater concern than packet delay. For a session using small windows, however, the round robin method guarantees a small cross-network delay for each packet while still guaranteeing a certain minimum service rate. † This service rate determines the maximum session throughput rate that can be supported and also roughly determines, for a given throughput rate, the delay of packets waiting to be admitted to the network. (A session requiring a larger guaranteed service rate and/or a smaller guaranteed cross-network delay could be visited more than once in each polling cycle.) ‡ This performance is

---

† In order for these guarantees to be meaningful, the routing algorithm should limit the number of sessions sharing any link and the number of hops in any session's path.

‡ While reducing a session's window size should decrease the cross-network delay of that session's packets, it might also increase the time the packets spend waiting to enter the network. Thus the total delay (i.e., pre-network plus cross-network) could actually increase as the window size decreases. (Mukherji [21] shows this for a different flow control scheme.) Small windows do offer certain advantages to the session, however. The session becomes aware of congestion problems earlier and can respond to large pre-network queues by compressing its data, prioritizing its data (and deferring

suited to interactive data, packet voice, and other applications requiring low packet delays. These guarantees for sessions with small windows apply even if other sessions in the network are using larger windows. Hence this scheme should be well suited to integrated services networks, i.e., those supporting a variety of applications. Delay-sensitive sessions can use small windows to meet their needs, and the remaining transmission capacity can be fairly divided among the other sessions by assigning them large windows.

It should be mentioned that the performance of the round robin method improves as the packet size (used by all sessions) is decreased. If the window sizes -- measured in *packets* -- are fixed, then the cross-network delay (measured in seconds per packet) drops as the packet size is reduced. If the window sizes -- measured in *bits* -- are fixed, then the throughput fairness improves as the packet size is reduced. Of course, these beneficial effects are balanced by the fact that packet overhead is more significant for small packet sizes.

This thesis shows that round robin scheduling with windows compares favorably to first-come-first-served scheduling with windows. † This finding is

---

or discarding the low priority items), or requesting a higher service rate from the network. Moreover, the component of total delay that *cannot* be directly observed or controlled by the session, viz., the cross-network component, is *guaranteed* to be small if small windows are used.

† In modeling first-come-first-served scheduling, this thesis assumes that link-by-link windows are used and that a packet may not join a link transmitter queue until it obtains a window permit for its next buffer. Perhaps a

of practical interest, since window flow control is commonly used. The max-min fairness results for round robin scheduling with large windows do not apply for first-come-first-served scheduling. Even when large windows are used, the session throughput rates in a first-come-first-served system depend strongly on the relative window sizes of competing sessions and the initial conditions of the . When small windows are used, the throughput and delay guarantees for round robin systems are also better than those for first-come-first-served systems. (This is not to say that round robin scheduling performs better in every case. I have seen systems where first-come-first-served scheduling produces slightly fairer throughputs.) Moreover, these first-come-first-served guarantees for a session x depend on the window sizes of the other sessions, whereas the round robin guarantees depend only on x's window size. Hence round robin scheduling simplifies the problem of selecting window sizes.

A simplistic description of the capacity allocation mechanisms of these two disciplines may help explain why their throughput performance is so different. First-come-first-served scheduling allocates link capacity to sessions according to the *average number* of packets each session has waiting. Round robin scheduling, on the other hand, considers the *fraction of time* each session has *at least one* packet waiting.

---

different implementation of first-come-first-served scheduling would perform better.

This thesis assumed that the propagation delays of the network links were negligible. The difference in worst-case throughput performance for round-robin and first-come-first-served scheduling should be even more pronounced if propagation delays are significant. Consider a system that uses first-come-first-served scheduling and has the layout shown in Figure 13. Session x uses links l_1 and l_2. Sessions $y_1, y_2, \ldots, y_{N-1}$ also use l_2. Flow control permits for the sessions are returned over links l_3 and l_4, whose directions are opposite to l_1 and l_2, respectively. Suppose that the round trip propagation delay over links l_1 and l_3 is d times the length of a packet transmission slot, while the propagation delay over the other links is negligible. Suppose that the window size for every session except x is w packets. If sessions $x, y_1, \ldots, y_{k-1}$ have very high demand and sessions $y_k, \ldots, y_{N-1}$ have very low demand, and if each active session is to receive its fair throughput of $\frac{1}{k}$ packets per slot, then $W(x,2)$ should be roughly $\frac{d}{k} + w$ packets. The fair window size for one value of k is unfair for other values, and the inequity worsens as the propagation delay grows. I conjecture that this problem is much less severe if round robin scheduling is used instead of first-come-first-served scheduling.

A similar difficulty arises if end-to-end windows are used instead of link-by-link windows. Consider a system that uses first-come-first-served scheduling with end-to-end windows and has the layout shown in Figure 14. Session x uses links l_1, l_2 and l_3 and shares each link l_i with single-hop

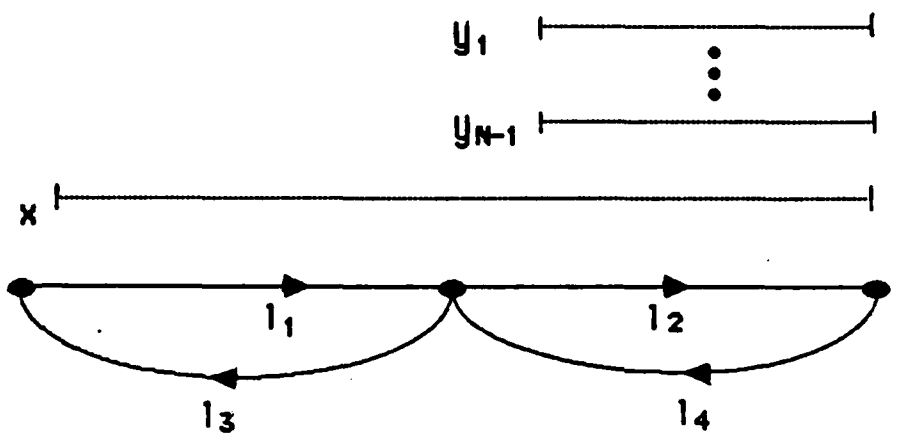


FIGURE 13

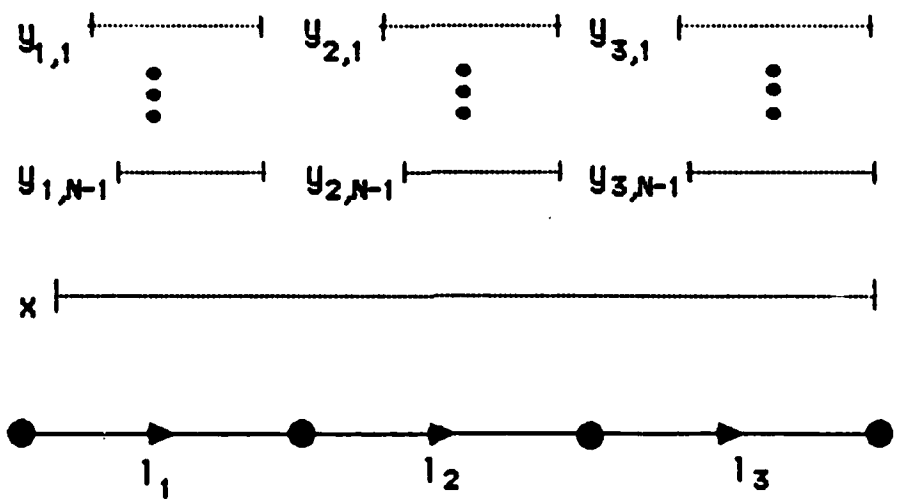


FIGURE 14

sessions $y_{i,1}$, $y_{i,2}$, ... , $y_{i,N-1}$. (For each of these links, there is another link with opposite direction that is not shown in Figure 14 and is used only to return flow control permits. The propagation delays of all links are negligible.) Suppose that N is very large and that the window size for every session except x is w packets. If all sessions have heavy demand, then session x's window size should be approximately $3w$ packets in order to achieve its fair throughput rate of $1/N$ packets per slot. If sessions $y_{1,1}$, ... , $y_{1,N-1}$, $y_{2,1}$, ... , $y_{2,N-1}$ have very low demand, however, and sessions x, $y_{3,1}$, ... , $y_{3,N-1}$ have high demand, then x's window size should only be about w packets. Window sizes that are fair for one scenario are unfair for the other, and the inequity worsens as the path length grows. I conjecture that this problem with end-to-end windows is much less severe if round robin scheduling is used instead of first-come-first-served scheduling.

For a final comparison of these two disciplines, consider a system with link-by-link windows. Suppose one session adjusts its window size to optimize its own throughput and delay. Suppose that this session has a great many competitors with large windows. I conjecture that the network appears very different to such a session when round robin scheduling is used than when first-come-first-served scheduling is used. I suspect that with round robin scheduling the session can vary its delay over a very wide range and can vary its throughput between zero and the max-min fair rate. With first-come-first-served scheduling, however, I suspect that the session can vary its throughput

over a very wide range, even taking unfair fractions of link capacity, but can hardly control its delay at all. The round robin scenario seems preferable to me. These first-come-first-served problems are compounded if *all* the sessions are adjusting their window sizes. A session primarily interested in maximizing its throughput will increase its window size. If its competitors follow suit, no one will get any more throughput and everyone's cross-network delay will increase. Eventually the delay increases may halt this game, but I believe that round robin scheduling would produce an earlier truce.

The costs and benefits of the round robin strategy can also be compared with the other fair flow control schemes mentioned in Section 1.1. Bially, Gold, and Seneff [2], Jaffe [16, 17], Gerla and Staskauskas [11: Section 3], Hayden [13: Chapters 4-5], Gafni [5: Chapter 3], Gafni and Bertsekas [6], Oshinsky [22], and Mosely [20] propose algorithms and session source controls to achieve max-min fair rates. Since these schemes actually compute the max-min fair rates, they can accomodate more variations in the objective function than are possible with the round robin strategy. For one thing, link delays can be considered in the optimization. The round robin method's provisions for delay management, viz., window size adjustment and polling frequency adjustment, are less systematic but are more suitable when different sessions have different delay requirements. To compare the costs of round robin and the other approaches, the computation and communication overhead of the max-min algorithms plus the difficulty of source rate control (i.e., variable rate

vocoding or variable packet sizing or packet metering) must be weighed against the accounting burden of managing round robin schedules and windows plus the cost of the link capacity needed to transmit window permits. The round robin method compares more favorably if session demands change frequently or if session lifetimes are short, since the max-min algorithms must be executed often under those circumstances.

The round robin strategy should also be compared with the approaches of Golestaani and Gallager [12, 8], Gerla and Staskauskas [11: Section 5.2], Thaker and Cain [26], Ibe [14], Gafni [5: Chapters 4-6], and Sauve, Wong and Field [24, 25]. The round robin method has less overhead, because neither the target session rates nor the window sizes nor the schedule parameters need to be computed, but round robin is limited to a smaller variety of throughput objectives. While Golestaani and Gallager, Gerla and Staskauskas, and Thaker and Cain also include in their objectives the cross-network delay averaged over all sessions, they cannot easily accomodate individual sessions with stringent delay requirements. The round robin scheme is better at this. Sauve, Wong and Field (in a related paper [27]), Ibe, and Gafni solve this problem by using various forms of priority queuing. The overhead costs of window flow control apply to all the strategies of this paragraph. A round robin schedule itself may be slightly harder to execute than the first-come-first-served schedules apparently assumed by Golestaani and Gallager, Gerla and Staskauskas, and Thaker and Cain, but it is no more difficult than the priority scheduling of Ibe

and Gafni, and it is much easier than the schedules of Sauve, Wong and Field, which depend on each packet's real-time delay.

Mukherji's flow control strategy [21] is extremely flexible. By correctly setting periodic link schedules, virtually any feasible set of session throughput rates can be enforced, with very small cross-network delays. The difficulty, of course, is in determining the desired set of throughputs. Optimizing the fairness of the target throughputs or minimizing the delay built into the link schedules could incur overhead penalties the round robin method does not have. However, with almost no effort Mukherji can offer throughput and delay guarantees comparable to the small-window guarantees of the round robin method. In fact, since Mukherji recommends round robin re-scheduling of transmission slots not claimed by their rightful owners, the performance of the two strategies should be similar in many applications. The overhead for executing the schedules and enforcing the windows in the two schemes should be comparable as well. Mukherji's method also has the property (described earlier for round robin) that an individual session can choose its window size to suit its cross-network delay requirements.

This thesis, in its examples as well as its analysis, focused on the *worst-case* performance of round robin scheduling with windows. An important area for future study is the *typical* performance of the scheme. Of interest are the following items, as functions of the window size: the fairness of the session throughput rates, the burstiness of the session flows, the severity of transients

arising from the initiation and termination of sessions and from changes in session demands, and the cross-network delay. Unfortunately, since many of these performance measures are very sensitive to the network topology, the session routes and demand rates, and the initial conditions, such a study would likely require the simulation of a great many sample networks of at least moderate size. It would also be interesting to see whether the use of end-to-end windows instead of link-by-link windows significantly changes the performance of the strategy. Link models with propagation delays and unequal capacities could also be considered. Finally, it would be worthwhile to examine variations of this method to see if max-min fair throughput rates can be achieved without computing the rates but without incurring large cross-network delays. One approach is to dynamically adjust the window sizes so that they are no larger than necessary. Another possibility is to change the round robin discipline slightly, e.g., by randomly rearranging the polling order of the sessions from time to time. This might ensure that the system enters very unfair configurations only rarely and only for brief periods.

APPENDICES

This section contains three lemmas. Lemma 8 shows that if a certain type of function $G(s, t)$ is linear in $t-s$ to within given error bounds, then tighter error bounds apply for sufficiently large s and t. Lemma 9 bounds the probability that a Bernoulli process segment of a given length will have a given degree of smoothness. Lemma 10 uses the strong law of large numbers to compute the following limit:

$$\lim_{q \to \infty} \max \left[0, \frac{g_1}{q}, \frac{g_1 + g_2}{q}, \ldots, \frac{g_1 + g_2 + \cdots + g_q}{q} \right]$$

for independent, identically distributed random variables $g_1, g_2, \ldots$.

A.1 Lemma 8: Symmetry of Upper and Lower Bounds in Steady State

Let $g(\cdot)$ be a real function of an integer argument. Define a real function $G(\cdot,\cdot)$ with integer arguments as follows:

$$G(s, t) = \sum_{u=s+1}^{t} g(u) \tag{334}$$

(Note that $G(s, t) = 0$ for $s \geq t$.) Let T be some integer, and let r, f', and f'' be some real numbers. Suppose that, for all integers s and t satisfying $T \leq s \leq t$,

$$-f' \leq G(s, t) - r\cdot(t - s) \leq f'' \tag{335}$$

It follows that, for every positive real number ϵ, there exists an integer $T_\epsilon \geq T$ such that, for all integers s and t satisfying $T_\epsilon \leq s \leq t$,

$$-\min\left[f', (f'' + \epsilon)\right] \leq G(s, t) - r\cdot(t - s) \leq \min\left[f'', (f' + \epsilon)\right] \tag{336}$$

Proof of Lemma 8

Assume that

$$f' \leq f'' \tag{337}$$

(The proof for $f' > f''$ is similar and will not be presented.) Because of assumptions (337) and (335), most of (336) is trivially true. All that must be proved is this: for every positive real number ϵ, there exists an integer $T_\epsilon \geq T$ such that, for all integers s and t satisfying $T_\epsilon \leq s \leq t$,

(338) $$G(s, t) \leq r \cdot (t - s) + f' + \epsilon$$

The proof will be by contradiction. Suppose that there exists a positive real number ϵ such that, for all integers $\hat{T} \geq T$, there exist integers s and t such that $\hat{T} \leq s \leq t$ and such that

(339) $$G(s, t) > r \cdot (t - s) + f' + \epsilon \quad .$$

For any positive integer K, this supposition can be applied repeatedly to construct a sequence of integers $s_1, t_1, s_2, t_2, \ldots, s_K, t_K$ such that $T \leq s_1 \leq t_1 \leq s_2 \leq t_2 \leq \cdots \leq s_K \leq t_K$ and such that

(340) $$G(s_k, t_k) > r \cdot (t_k - s_k) + f' + \epsilon \qquad \text{for } k = 1, 2, \ldots, K$$

Summing (340) over k yields:

(341) $$\sum_{k=1}^{K} G(s_k, t_k) > r \cdot \sum_{k=1}^{K} (t_k - s_k) + K \cdot f' + K \cdot \epsilon$$

Now apply assumption (335) to $G(t_1, s_2), G(t_2, s_3), \ldots, G(t_{K-1}, s_K)$ and sum over k.

$$\sum_{k=1}^{K-1} G(t_k, s_{k+1}) \geq \sum_{k=1}^{K-1} \left[r \cdot (s_{k+1} - t_k) - f' \right]$$

(342) $$= r \cdot \sum_{k=1}^{K-1} (s_{k+1} - t_k) - (K - 1) \cdot f'$$

(Note that (342) holds even if $K = 1$.) Definition (334) can be used to add (341) and (342) together.

$$G(s_1 , t_K) = \sum_{u=s_1+1}^{t_K} g(u)$$

$$= \left[\sum_{k=1}^{K} \sum_{u=s_k+1}^{t_k} g(u)\right] + \left[\sum_{k=1}^{K-1} \sum_{u=t_k+1}^{s_{k+1}} g(u)\right]$$

$$= \left[\sum_{k=1}^{K} G(s_k , t_k)\right] + \left[\sum_{k=1}^{K-1} G(t_k , s_{k+1})\right]$$

$$> \left[r \cdot \sum_{k=1}^{K} (t_k - s_k) + K \cdot f' + K \cdot \epsilon\right]$$

$$+ \left[r \cdot \sum_{k=1}^{K-1} (s_{k+1} - t_k) - (K-1) \cdot f'\right]$$

(343) $$= r \cdot (t_K - s_1) + K \cdot \epsilon + f'$$

For sufficiently large K, viz.:

(344) $$K > \frac{f'' - f'}{\epsilon}$$

relation (343) contradicts assumption (335). This completes the proof of Lemma 8.

A.2 Lemma 9: Smoothness of a Bernoulli Process

Let $g(t)$ be a Bernoulli process with rate μ, $0 \leq \mu \leq 1$. Let $G(s, t)$ be the number of successes among $g(s+1)$, $g(s+2)$, ... , $g(t)$; note that $G(s, t) = 0$ if $s \geq t$. For any positive real number Δ, any integer T, and any positive integer τ, it follows that

$$(345) \qquad \text{PROB} \left\{ \begin{array}{c} |G(s, t) - \mu \cdot (t - s)| \leq \Delta \\ \text{for all } s, t \text{ such that } T \leq s < t \leq T+\tau \end{array} \right\} \geq 1 - \frac{\tau}{\Delta^2}$$

Proof of Lemma 9

Let s and t be any integers such that $T \leq s < t \leq T+\tau$. Note that

$$(346) \qquad G(s, t) = G(T, t) - G(T, s)$$

Consequently, if

$$|G(T, s) - \mu \cdot (s - T)| \leq \frac{\Delta}{2}$$

and

$$|G(T, t) - \mu \cdot (t - T)| \leq \frac{\Delta}{2}$$

then

$$|G(s, t) - \mu \cdot (t - s)| \leq \Delta$$

Therefore, to prove (345) it suffices to show that

$$(347) \qquad \text{PROB} \left\{ \begin{array}{l} |\, G(T, u) - \mu \cdot (u - T)| \leq \frac{\Delta}{2} \\ \text{for all } u \text{ such that } T < u \leq T+\tau \end{array} \right\} \geq 1 - \frac{\tau}{\Delta^2}$$

This is the same as proving that

$$(348) \qquad \text{PROB} \left\{ \begin{array}{l} |\sum_{k=1}^{K} G(T+k-1, T+k) - \mu \cdot K| \leq \frac{\Delta}{2} \\ \text{for all } K \text{ such that } 1 \leq K \leq \tau \end{array} \right\} \geq 1 - \frac{\tau}{\Delta^2}$$

Inequality (348) follows from Kolmogorov's inequality [3] and the fact that

$$\textit{VARIANCE} \{ G(T, T+\tau) \} = \mu \cdot (1-\mu) \cdot \tau$$

$$(349) \qquad \leq \frac{\tau}{4}$$

This completes the proof of Lemma 9.

A.3 Lemma 10: A Corollary of the Strong Law of Large Numbers

Suppose that $g_1, g_2, \ldots$ are independent, identically distributed random variables with mean μ. For all integers K and q such that $0 \leq K \leq q$, define G_q^K and G_q as follows:

$$G_q^K = \frac{1}{q} \cdot \sum_{k=1}^{K} g_k \tag{350}$$

$$G_q = \max_{0 \leq K \leq q} G_q^K \tag{351}$$

It follows that, with probability one,

$$G_q \xrightarrow[q \to \infty]{} \max[0, \mu] \tag{352}$$

Proof of Lemma 10

It follows from (350) and the strong law of large numbers [3] that, with probability one,

$$G_K^K \xrightarrow[K \to \infty]{} \mu \tag{353}$$

This means that there exists a subset Γ of the sample space such that

$$PROB\{\Gamma\} = 1 \tag{354}$$

and such that, for every sample point γ in Γ and every positive real number ϵ, there exists a positive integer $Q(\gamma, \epsilon)$ such that

(355) $$| G_K^K(\gamma) - \mu | \leq \epsilon \qquad \text{for all } K \geq Q(\gamma, \epsilon)$$

To prove (352), it will be shown that for every γ in Γ and every positive real number ϵ, there exists a positive integer $Q'(\gamma, \epsilon)$ such that

$$| G_q(\gamma) - \max[0, \mu] | \leq \epsilon \qquad \text{for all } q \geq Q'(\gamma, \epsilon)$$

Let γ be any sample point in Γ and let ϵ be any positive real number. Define $Q'(\gamma, \epsilon)$ as follows.

(356) $$Q'(\gamma, \epsilon) = \max \left[Q(\gamma, \epsilon), \left[\frac{1}{\epsilon} \cdot \sum_{k=1}^{Q(\gamma, \epsilon)} | g_k(\gamma) | \right] \right]$$

Let q be any integer such that

(357) $$q \geq Q'(\gamma, \epsilon)$$

The goal is to show that

(358) $$| G_q(\gamma) - \max[0, \mu] | \leq \epsilon$$

First, let us find a lower bound for $G_q(\gamma)$. It follows from (351) and (350) that

$$G_q(\gamma) = \max_{0 \le K \le q} G_q^K(\gamma)$$

$$\geq G_q^0(\gamma)$$

$$= 0$$

$$\geq -\epsilon \tag{359}$$

Using (351), (355), (357), and (356), a different lower bound can be found:

$$G_q(\gamma) = \max_{0 \le K \le q} G_q^K(\gamma)$$

$$\geq G_q^q(\gamma)$$

$$\geq \mu - \epsilon \tag{360}$$

Combining (359) and (360) yields:

$$G_q(\gamma) \geq \max[0, \mu] - \epsilon \tag{361}$$

This is half of the desired result (358).

Now an upper bound can be found for $G_q(\gamma)$ by proving the following bound for $G_q^K(\gamma)$.

$$G_q^K(\gamma) \leq \max[0, \mu] + \epsilon \qquad \text{for } K = 0, 1, \ldots, q \tag{362}$$

The small and large values of K must be treated separately. For $0 \leq K \leq Q(\gamma, \epsilon)$, apply (350), (357), and (356):

$$G_q^K(\gamma) = \frac{1}{q} \cdot \sum_{k=1}^{K} g_k(\gamma)$$

$$\leq \frac{1}{q} \cdot \sum_{k=1}^{K} |g_k(\gamma)|$$

$$\leq \frac{1}{q} \cdot \sum_{k=1}^{Q(\gamma, \epsilon)} |g_k(\gamma)|$$

$$\leq \frac{1}{Q'(\gamma, \epsilon)} \cdot \sum_{k=1}^{Q(\gamma, \epsilon)} |g_k(\gamma)|$$

$$\leq \epsilon$$

$$\leq \max[0, \mu] + \epsilon$$

For $Q(\gamma, \epsilon) \leq K \leq q$, apply (350) and (355):

$$G_q^K(\gamma) = \frac{1}{q} \cdot \sum_{k=1}^{K} g_k(\gamma)$$

$$= \frac{K}{q} \cdot \frac{1}{K} \cdot \sum_{k=1}^{K} g_k(\gamma)$$

$$= \frac{K}{q} \cdot G_K^K(\gamma)$$

$$\leq \frac{K}{q} \cdot (\mu + \epsilon)$$

$$\leq \frac{K}{q} \cdot (\max[0, \mu] + \epsilon)$$

$$\leq \max[0, \mu] + \epsilon$$

This completes the proof of (362). From (351) and (362), it follows that

$$G_q(\gamma) = \max_{0 \leq K \leq q} G_q^K(\gamma)$$

$$\leq \max[0, \mu] + \epsilon \tag{363}$$

This is the second half of the desired result (358), completing the proof of Lemma 10.

REFERENCES

[1] Bharath-Kumar, K. and J. M. Jaffe, "A New Approach to Performance-Oriented Flow Control," *IEEE Trans. Comm.*, Vol. COM-29, No. 4, April 1981, pp. 427-435.

[2] Bially, T., B. Gold, and S. Seneff, "A Technique for Adaptive Voice Flow Control in Integrated Packet Networks," *IEEE Trans. Comm.*, Vol. COM-28, No. 3, March 1980, pp. 325-333.

[3] Billingsley, P., *Probability and Measure*, Wiley, N. Y., N. Y., 1986.

[4] Eilon, S., "A Simpler Proof of $L=\lambda W$," *Operations Research*, Vol. 17, No. 5, Sept.-Oct. 1969, pp. 915-917.

[5] Gafni, E. M., "The Integration of Routing and Flow-Control for Voice and Data in a Computer Communication Network," Report LIDS-TH-1239, Lab. for Info. and Decision Sys., Mass. Inst. of Technology, Cambridge, Mass., Sept. 1982.

[6] Gafni, E. M. and D. P. Bertsekas, "Dynamic Control of Session Input Rates in Communication Networks," *IEEE Trans. Auto. Control*, Vol. AC-29, No. 11, Nov. 1984, pp. 1009-1016.

[7] Gallager, R. G., D. P. Bertsekas, and P. A. Humblet, "Data Networks Reliability," Research Proposal to the Defense Advanced Research Projects Agency, 1983.

[8] Gallager, R. G. and S. J. Golestaani, "Flow Control and Routing Algorithms for Data Networks," *Proc. Fifth Internatl. Conf. Comp. Comm.*, Oct. 1980, pp. 779-784.

[9] Gallager, R. G. and P. A. Humblet, "The Dynamics of Data Network Research," Research Proposal to the National Science Foundation, 1983.

[10] Gerla, M. and L. Kleinrock, "Flow Control: A Comparative Survey," *IEEE Trans. Comm.*, Vol. COM-28, No. 4, April 1980, pp. 553-574.

[11] Gerla, M. and M. Staskauskas, "Fairness in Flow Controlled Networks," *Proc. IEEE Internatl. Conf. Comm.*, June 1981, pp. 63.2.1-63.2.5.

[12] Golestaani, S. J., "A Unified Theory of Flow Control and Routing in Data Communication Networks," Report LIDS-TH-963, Lab. for Info.

and Decision Sys., Mass. Inst. of Technology, Cambridge, Mass., Jan. 1980.

[13] Hayden, H. P., "Voice Flow Control in Integrated Packet Networks," Report LIDS-TH-1152, Lab. for Info. and Decision Sys., Mass. Inst. of Technology, Cambridge, Mass., Oct. 1981.

[14] Ibe, O. C., "Flow Control and Routing in an Integrated Voice and Data Communication Network," Report LIDS-TH-1115, Lab. for Info. and Decision Sys., Mass. Inst. of Technology, Cambridge, Mass., August 1981.

[15] Iosifescu, M., *Finite Markov Processes and Their Applications*, Wiley, N.Y., N.Y., 1980.

[16] Jaffe, J. M., "A Decentralized, 'Optimal,' Multiple-User, Flow Control Algorithm," *Proc. Fifth Internatl. Conf. Comp. Comm.*, Oct. 1980, pp. 839-844.

[17] Jaffe, J. M., "Bottleneck Flow Control," *IEEE Trans. Comm.*, Vol. COM-29, No. 7, July 1981, pp. 954-962.

[18] Jaffe, J. M., "Flow Control Power is Nondecentralizable," *IEEE Trans. Comm.*, Vol. COM-29, No. 9, Sept. 1981, pp. 1301-1306.

[19] Little, J. D. C., "A Proof of the Queueing Formula $L=\lambda W$," *Operations Research*, Vol. 9, No. 3, May-June 1961, pp. 383-387.

[20] Mosely, J., "Asynchronous Distributed Flow Control Algorithms," Report LIDS-TH-1415, Lab. for Info. and Decision Sys., Mass. Inst. of Technology, Cambridge, Mass., Oct. 1984.

[21] Mukherji, U., "A Schedule-Based Approach for Flow-Control in Data Communication Networks," Report LIDS-TH-1527, Lab. for Info. and Decision Sys., Mass. Inst. of Technology, Cambridge, Mass., Jan. 1986.

[22] Oshinsky, D. A., "Use of Fair Rate Assignment Algorithms in Networks with Bursty Sessions," S. M. Thesis, Dept. of Elec. Engr. and Comp. Sci., Mass. Inst. of Technology, Cambridge, Mass., May 1984.

[23] Regnier, J. M., "Priority Assignment in Integrated Services Networks," Report LIDS-TH-1565, Lab. for Info. and Decision Sys., Mass. Inst. of Technology, Cambridge, Mass., May 1986.

[24] Sauve, J. P., J. W. Wong, and J. A. Field, "On Throughput and Fairness in Packet Switching Networks with Window Flow Control," CCNG Report E-100, Computer Communications Networks Group, Univ. of Waterloo, Waterloo, Ontario, Canada, Dec. 1981.

[25] Sauve, J. P., J. W. Wong, and J. A. Field, "Improving Total Throughput in Packet Switching Networks with Window Flow Control," *Proc. IEEE Global Telecomm. Conf.*, Nov.-Dec. 1982, pp. 1189-1194.

[26] Thaker, G. H. and J. B. Cain, "Interactions Between Routing and Flow Control Algorithms," *IEEE Trans. Comm.*, Vol. COM-34, No. 3, March 1986, pp. 269-277.

[27] Wong, J. W., J. P. Sauve, and J. A. Field, "A Study of Fairness in Packet-Switching Networks," *IEEE Trans. Comm.*, Vol. COM-30, No. 2, Feb. 1982, pp. 346-353.

GLOSSARY OF NOTATION

SYMBOL	MEANING	DEFINED IN
$[s, t]$	Time interval from beginning of slot s to end of slot t; null if $s > t$	§ 2.1
$(s, t]$	Same as $[s+1, t]$	§ 2.1
$[s, t)$	Same as $[s, t-1]$	§ 2.1
(s, t)	Same as $[s+1, t-1]$	§ 2.1
S	Number of sessions in network	§ 2.2
$N'(l)$	Number of sessions using link l	§ 2.2
$N(x, h)$	Number of sessions using hop h of session x	§ 2.2
$N(x)$	Maximum number of sessions using any link in path of session x	§ 2.2
N	Maximum number of sessions sharing any link in network	§ 2.2
$H(x)$	Number of links in path of session x	§ 2.2
H	Maximum number of links in path of any session in network	§ 2.2

$W(x, h)$	Window size (capacity) for buffer h of session x	§ 2.3
W'	Maximum window size for any session x at any buffer h in the range $1 \leq h \leq H(x)$	§ 2.3
W''	Maximum window size for any session x at any buffer h in the range $2 \leq h \leq H(x)+1$	§ 2.3
$B(x, h, t)$	Level of buffer h of session x at time t	§ 2.2
T_{SS}	Time when throughputs and buffer levels stabilize	§ 4.4
$m(x, h)$	Minimum level of buffer h of session x after time T_{SS}	§ 4.4
$M(x, h)$	Maximum level of buffer h of session x after time T_{SS}	§ 4.4
$P(x, h, s, t)$	Throughput (number of transmitted packets) for session x over hop h during interval $(s, t]$	§ 2.2
$P'(x, l, s, t)$	Throughput (number of transmitted packets) for session x over link l during interval $(s, t]$	§ 2.2

$R_A(x)$	Long-term average throughput of session x	§ 2.2
$\Upsilon(x, h, p)$	Time slot during which session x transmits packet p over hop h	§ 2.2
$\Xi(x, p)$	Cross-network delay for packet p of session x	§ 2.2
$C(x, h, s, t)$	Number of chances for session x to transmit over hop h during interval $(s, t]$; $C(x, 0, s, t)$ is number of attempts by x to insert packets into buffer 1 during $(s, t]$	§ 2.4.1; § 2.5
$C'(x, l, s, t)$	Number of chances for session x to transmit over link l during interval $(s, t]$	§ 2.4.1
$A'(l)$	Schedule delay bound for link l	§ 2.4.3
$A(x, h)$	Schedule delay bound for hop h of session x	§ 2.4.3
$A(x)$	Maximum schedule delay bound for any link in path of session x	§ 2.4.3
Ω	Sample space consisting of all possible sample paths of system demand	§ 2.5
$\lambda(x)$	Demand rate of session x	§ 2.5

$R_F(i)$	i^{th} smallest distinct fair rate; fair rate for any session with congestion index i	§ 4.1
$I(x)$	Congestion index of session x	§ 4.1
I	Number of distinct fair rates; maximum congestion index of any session	§ 4.1
$R_C(x, h)$	Lower bound on fair rate of chances for session x at hop h	§ 4.1
$R'_C(x, l)$	Lower bound on fair rate of chances for session x at link l	§ 4.1
D_{CL}, E_{CL}, F_{CL}	Functions used as lower bounds on numbers of chances received by sessions	§ 4.2
D_{PL}, E_{PL}, F_{PL}	Functions used as lower bounds on numbers of packets transmitted by sessions	§ 4.2
$D_{PU}, E_{PU}, F_{PU}, F''_{PU}$	Functions used as upper bounds on numbers of packets transmitted by sessions	§ 4.2

Distribution List

Defense Documentation Center Cameron Station Alexandria, Virginia 22314	12 Copies
Assistant Chief for Technology Office of Naval Research, Code 200 Arlington, Virginia 22217	1 Copy
Office of Naval Research Information Systems Program Code 437 Arlington, Virginia 22217	2 Copies
Office of Naval Research Branch Office, Boston 495 Summer Street Boston, Massachusetts 02210	1 Copy
Office of Naval Research Branch Office, Chicago 536 South Clark Street Chicago, Illinois 60605	1 Copy
Office of Naval Research Branch Office, Pasadena 1030 East Greet Street Pasadena, California 91106	1 Copy
Naval Research Laboratory Technical Information Division, Code 2627 Washington, D.C. 20375	6 Copies
Dr. A. L. Slafkosky Scientific Advisor Commandant of the Marine Corps (Code RD-1) Washington, D.C. 20380	1 Copy

Office of Naval Research
Code 455
Arlington, Virginia 22217

1 Copy

Office of Naval Research
Code 458
Arlington, Virginia 22217

1 Copy

Mr. E. H. Gleissner
Naval Ship Research & Development Center
Computation and Mathematics Department
Bethesda, Maryland 20084

1 Copy

Captain Grace M. Hopper
Naval Data Automation Command
Code 00H
Washington Navy Yard
Washington, DC 20374

1 Copy

Advanced Research Projects Agency
Information Processing Techniques
1400 Wilson Boulevard
Arlington, Virginia 22209

1 Copy

Dr. Stuart L. Brodsky
Office of Naval Research
Code 432
Arlington, Virginia 22217

1 Copy

Prof. Fouad A. Tobagi
Computer Systems Laboratory
Stanford Electronics Laboratories
Department of Electrical Engineering
Stanford University
Stanford, CA 94305

END

3-87

DTIC